C. 4

SANDIA REPORT

SAND97-8704 • UC–411
Unlimited Release
Printed October 1996

Modeling of Precipitation in Al Alloys

M. Asta, S. M. Foiles, W. G. Wolfer, T. P. Swiler, C. L. Rohrer,
R. W. Hyland, Jr., J. L. Murray, D. P. Ziegler, B. O. Hall, and K. Lippert

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94551
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release; distribution is unlimited.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.
NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of the contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors or subcontractors.

This report has been reproduced from the best available copy.

Available to DOE and DOE contractors from:

Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge TN 37831

Prices available from (615) 576-8401, FTS 626-8401.

Available to the public from:

National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.
Springfield, VA 22161

SAND97-8704 UC-411
Unlimited Release
Printed October 1996

MODELING OF PRECIPITATION IN Al ALLOYS

M. Asta, S. M. Foiles and W. G. Wolfer
Division 8717
Sandia National Laboratories / California

T. P. Swiler
Division 1841
Sandia National Laboratories / New Mexico

C. L. Rohrer, R. W. Hyland, Jr., J. L. Murray, D. P. Ziegler, B. O. Hall and K. Lippert
Alcoa Technical Center, Pennsylvania

ABSTRACT

In this report we summarize the results of a cooperative-research-and-development-agreement (CRADA) project, the main objective of which was the development of a computational model of precipitation from a supersaturated alloy solid solution. The model is based upon the formalism of chemical-reaction-rate theory, combined with classical descriptions of precipitate thermodynamic properties and a mean-field treatment of diffusion-limited growth and coarsening. For the specific case of precipitation of Al_3Sc in supersaturated Al-Sc alloys, we demonstrate how the computational model can be employed in order to calculate number densities and size distributions of precipitates as a function of aging time and temperature, including the effects of continuous cooling and thermally generated point defects. The application of our model to a specific alloy system requires the knowledge of diffusion data, point defect energetics, and thermodynamic properties for bulk phases and interphase interfaces. For interfaces and point defects thermodynamic data can be difficult to measure experimentally and reliable values of defect free energies are often unavailable. For this reason part of our efforts in the CRADA project were devoted to applying semiempirical and first-principles atomistic techniques to the calculation of interfacial and point-defect thermodynamic properties. In this report applications are discussed for interphase interfaces in the Al-Ag, Al-Sc and Al-Li alloy systems. We also describe atomistic work aimed at understanding the energetics of vacancy clusters in Al. These clusters serve as sinks for isolated vacancies during aging and their growth can lead to more complex defects, such as dislocation loops, that act as heterogeneous nucleation sites.

CONTENTS

MODELING OF PRECIPITATION IN Al ALLOYS

I. Summary

In this section an overall summary is given describing the CRADA project and its two main components: the rate-theory-based approach for modeling precipitation kinetics, and the atomistic methods for calculating thermodynamic properties. This section is followed by four others which give details of the computational methods and which contain results for specific applications.

Overview of CRADA Project

In Figure 1 a flow chart is shown which illustrates schematically the goals and different approaches involved in the CRADA project. The main goal of the project was to provide microstructural information which could be used in continuum models of mechanical properties in Al alloys. Specifically, the aim was to calculate precipitate size distributions as a function of chemical composition and thermal history. The microstructural information was calculated using a kinetic modeling code based upon chemical reaction-rate theory [1]. The theory provides a formalism for treating nucleation, growth and coarsening within a single, unified framework, and it is felxible enough to treat multiple phases and solute species. The input to the rate-theory computer code consists of a variety of thermochemical data including diffusion constants, elastic constants, bulk and interphase interfacial thermodynamic properties and point-defect energetics. Where they are available, the values of these parameters are taken from the Alcoa experimental database. In some cases, reliable values of important thermochemical parameters are unavailable. In these situations atomistic computational techniques provide valuable tools which can be used to calculate relevant properties. Our efforts in the CRADA project were divided into two areas: development and coding of the rate-theory for modeling precipitation kinetics, and application of atomistic techniques to the calculation of alloy thermodynamic properties and defect energetics.

Accomplishments

At present we have succeeded in developing a rate-theory model applicable to the kinetics of coherent, homogeneously-nucleated precipitates in a binary alloy. In the next section details of our model are given and results are presented for the application of Al_3Sc precipitation in supersaturated, Al-rich, Al-Sc alloys. The text of the next section is taken from Reference [2].

The results presented below for Al-Sc demonstrate the ability of the rate-theory model to describe simultaneously nucleation, growth, and size-coarsening kinetics. Additionally it is discussed below how the model can be used to study the ways in which transformation kinetics are affected by varied thermal processing paths and atomic-assembly mechanisms. A copy of the code used in the Al-Sc simulations is given at the end of the report in the Appendix.

In parallel with the rate-theory modeling of Al_3Sc precipitation kinetics, atomistic calculations were undertaken [3] which were aimed at computing values of the interfacial excess free energy for the Al/Al_3Sc interphase boundary. In the third section we present results for Al/Al_3Sc thermodynamic properties, calculated using the embedded-atom-method (EAM) [4] in combination with Monte-Carlo simulations and a low-temperature expansion technique.

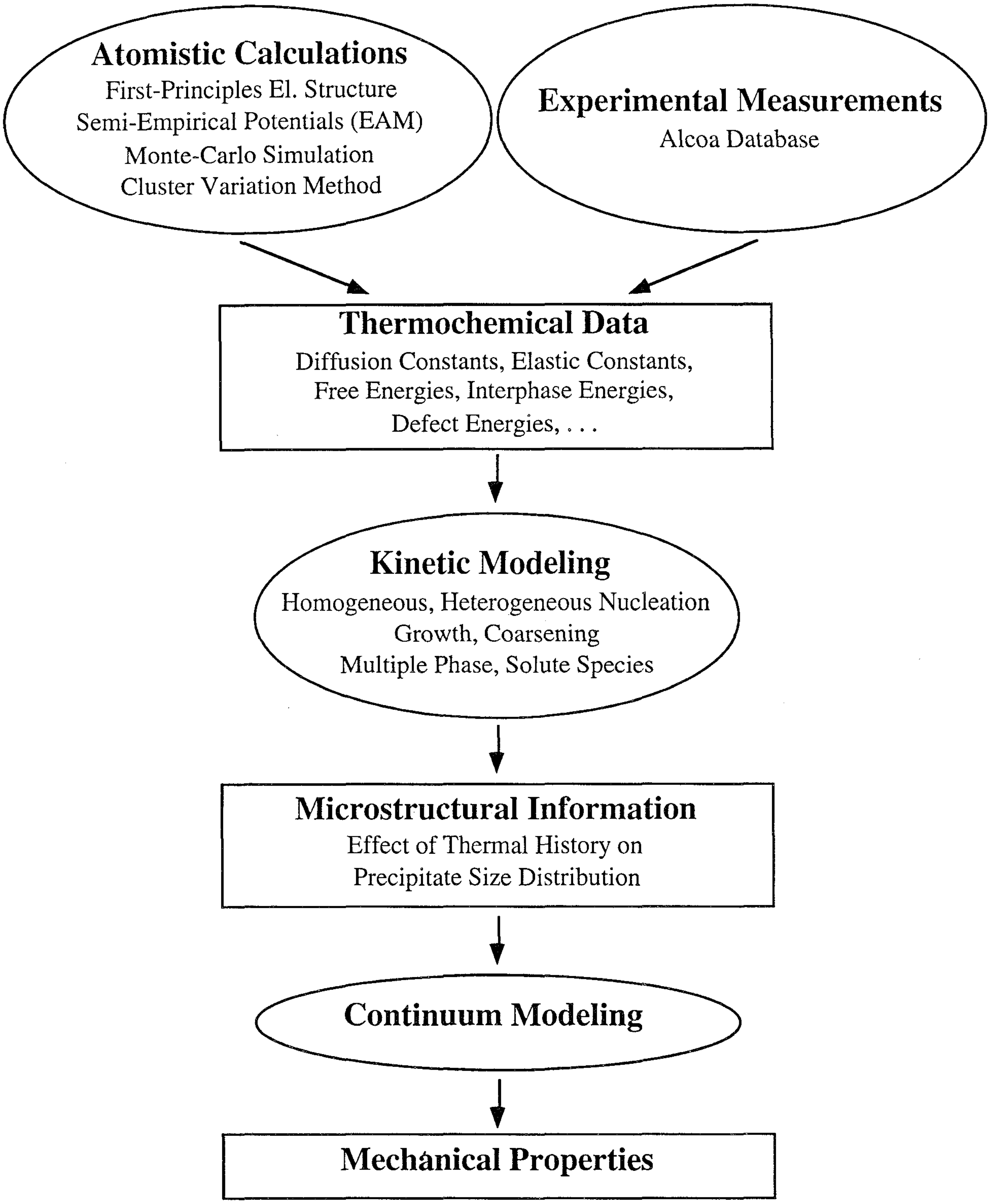

Atomistic Calculations
First-Principles El. Structure
Semi-Empirical Potentials (EAM)
Monte-Carlo Simulation
Cluster Variation Method
Experimental Measurements
Alcoa Database
Thermochemical Data
Diffusion Constants, Elastic Constants,
Free Energies, Interphase Energies,
Defect Energies, . . .
Kinetic Modeling
Homogeneous, Heterogeneous Nucleation
Growth, Coarsening
Multiple Phase, Solute Species
Microstructural Information
Effect of Thermal History on
Precipitate Size Distribution
Continuum Modeling
Mechanical Properties

Figure 1. Schematic illustration of the goals and different approaches involved in the CRADA.

One objective of the EAM work was to test the accuracy of the semi-empirical approach in its application to the Al-Sc system. It was found that the values of Al/Al_3Sc interfacial energies predicted by the EAM were significantly smaller than those derived from experimental data [5]. A possible reason for this apparent inadequacy of the EAM approach is the lack of sufficent data to which the atomic potentials could be fit. In situations such as this where sufficient experimental results are not available to allow a semi-empirical approach to be applied with high accuracy, a useful alternative is provided by first-principles techniques based upon quantum-mechanical, electronic-structure calculations.

In the past twenty years techniques have been derived for calculating alloy thermodynamic properties from first-principles [6]. Until recently, however such techniqes were applied to the calculation of *bulk* thermodynamic properties only. As part of the CRADA project we performed the first applications of such an approach to the study of interfacial thermodynamic properties [7]. The first-principles approach is summarized and examples are provided in the fourth section. For interphase boundaries between an Al solid-solution and the ordered Al_3Li phase, as well as between Al and the Guinier-Preston zone phase in Al-Ag, it will be demonstrated how the cluster-variation-method [8] can be applied to the calculation of interfacial excess free energies and composition profiles using either empirical or first-principles-calculated energetic interaction parameters.

In section V we present results of an atomistic study of the energetics of open and collapsed vacancy clusters in aluminum. Vacancies play an important role in determining the kinetics of precipitation of second phases in metals. They both enhance solute diffusion and serve as building blocks for more complex defects, such as dislocation loops, that act as heterogeneous nucleation sites. In the work presented in section V, we sought to determine the stability of vacancy clusters that act as sinks for single vacancies in aluminum.

Future Work

In the latest stages of the CRADA project work focussed on the extension of the rate-theory code in order to include multiple solute species, the effects of solute-vacancy interactions, the formation of vacancy clusters, and the possibility of heterogeneous nucleation on grain boundaries. Preliminary results obtained with the extended code are not presented in this report. Work along these lines is the subject of continuing collaborative research between Alcoa and Sandia.

Work also is continuing on the development and application of first-principles approaches for calculating interfacial thermodynamic properties. At the present time applications are focussing on the Al/Al_3Sc system as well as on interfaces in the Al-Ti alloy system. In the future efforts will need to be devoted to the consideration of interfaces between alloy phases with different parent lattice structures, and issues relating to incoherent interfaces will need to be addressed.

II. Kinetic Model of Precipitate Evolution

In this section the kinetic model for simulating precipitation reactions as a function of thermal history is discussed. The approach is based on the well known chemical reaction rate theory [1] which has been used in the past to study the kinetics of a wide variety of microstructure evolution problems [e.g. 3]. The intent of the work described in this section is to utilize the flexibility of the rate theory formalism to simultaneously model the processes of nucleation, growth, and coarsening in a single framework, allowing for generalizable alloy energetics, nonisothermal conditions, and the impact of thermal defects on the precipitation kinetics.

Method

In describing the process of precipitation, the reaction rates (the rates at which precipitates form and evolve to larger sizes) depend on the operative microscopic mechanisms for atomic attachment to and detachment from developing product phase particles. Five simplifications were assumed in the present work to define quantitatively the rate equations and the reaction rate coefficients. 1) The mechanism for size evolution of the product phase was assumed to be exclusively via the attachment or emission of a monomer solute species and only monomers were considered to be mobile. 2) The flux of monomers to stable product phase particles was assumed to be isotropic and particles were assumed to be spherical. 3) Only binary alloys decomposing into stoichiometric product phases were considered. 4) Atomic rearrangements associated with forming the product phase were assumed to occur quickly relative to the solute arrival rate. 5) Consistent with classical nucleation theory [4], it was assumed that the fastest forming embryos are those following the lowest free energy path and that this path is reasonably defined by the bulk thermodynamic free energy of the product phase together with interfacial and strain energy contributions. Embryos which form along alternative energy paths were considered to have a negligible effect on the overall transformation rate.

Consistent with these simplifications, a general set of master equations describing the precipitate size evolution can be written as

$$\frac{d\,N(i>1,j)}{d\,t} = [\,g(i-1,j)\,N(1)\,N(i-1,j) - e(i,j)\,N(i,j)\,] - [\,g(i,j)\,N(1)\,N(i,j) - e(i+1,j)\,N(i+1,j)\,] \qquad \text{(II.1a)}$$

$$\frac{d\,N(1)}{d\,t} = -\sum_{j}\sum_{i=2}[\,g(i-1,j)\,N(1)\,N(i-1,j) - e(i,j)\,N(i,j)\,] - [\,g(1,j)\,N(1)\,N(1) - e(2,j)\,N(2,j)\,] \qquad \text{(II.1b)}$$

where N(i,j) is the number density (m^{-3}) of phase j particles containing i solute atoms plus the appropriate number of solvent atoms to maintain the stoichiometry of the specified phase. The rate coefficients, g(i,j) and e(i,j), indicate the rates of attachment and emission, respectively, of one monomer to or from a phase j particle of size i. The evolution of the monomer number density is written separately in Eq. II.1b because no emission is possible from a monomer and emission from every greater size class results in the formation of one monomer along with a particle of the original size minus one. The fact that monomer attachment and dimer dissociation are counted twice in Eq. II.1b arises because the break up of a dimer results in the creation of two monomers and, likewise, dimer formation requires two monomers.

The rate coefficients, g(i) and e(i) (with the phase designation omitted for clarity), were defined separately for precritical "embryos" and for stable product phase precipitates in the spirit of the classical theories of nucleation and growth. Precipitates were considered stable once they had exceeded the critical radius, r*, plus a thermal factor, δ, defined as in Refs. [11] and [12].

Particles with radii less than r* + δ were considered to be thermodynamically unstable with respect to the matrix phase and to evolve by local atomic jumps with an energy barrier equal to the free energy change upon transforming a size i embryo to an embryo of size i+1. The growth rates for embryos of size i>1 were defined similarly to the derivation of Kelton and coworkers [13]:

$$g(i) = \left[6\,(a/\sqrt{2})^{-2}\,D_0\,e^{-(E_a/kT)}\,v(1)\right]\left[\frac{A_{i+1}}{A_i}\right]$$
$$\times\ \exp\left[-\frac{(\Delta G_v + \Delta G_s)(V_{i+1}-V_i) + \gamma(A_{i+1}-A_i)}{kT}\right] \qquad \text{(III.2)}$$

where a is the composition dependent matrix lattice parameter and $a/\sqrt{2}$ is the atomic jump length for an fcc lattice. The solute diffusivity is represented by the diffusivity prefactor, D_o, the barrier to solute migration via the monovacancy mechanism, E_a, and the monovacancy concentration, v(1). (The evolution of v(1) will be discussed later in this section.) A_i and V_i are the surface area and volume, respectively, of a size i embryo. The volume free energy of transformation, the elastic strain energy, and the interfacial free energy are designated by ΔG_v, ΔG_s, and γ, respectively. k and T have their usual meanings. The first two factors in parentheses will be labeled D′ and A′(i+1) in subsequent equations.

The emission rates for the embryos were derived using the principle of detailed balance. Specifically, it is assumed (as in classical nucleation theory [10]) that the forward and reverse reaction rates are equal under the condition of "constrained" equilibrium where the distribution of embryos is that which minimizes the total free energy of the system subject to the constraint of a fixed monomer concentration. The detailed balance criterion can be expressed as

$$g(i,j)\ N(1)\ \tilde{N}(i,j) = e(i+1,j)\ \tilde{N}(i+1,j) \qquad \text{(II.3)}$$

where the demarcated number densities represent constrained equilibrium quantities. By making use of the expression relating $N(i,j)$ to the exponential of the product phase formation energy [10], Eq. II.3 can be rewritten as

$$\frac{e(i+1)}{g(i)} = N(1)\,\frac{\tilde{N}(i)}{\tilde{N}(i+1)} = N(1)\,\exp\left[\frac{(\Delta G_v + \Delta G_s)(V_{i+1} - V_i) + \gamma(A_{i+1} - A_i)}{kT}\right]. \qquad \text{(II.4)}$$

Therefore, the embryo emission rates for i>1 can be expressed as

$$e(i+1) = D'\,A'(i+1)\,N(1) \qquad \text{(II.5)}$$

Detailed balance between monomers and dimers was specially treated based upon point defect arguments [15]. Thus, g(1) and e(2) were defined as

$$g(1) = 2\,D'\,A'(2)$$

$$e(2) = g(1)\,\tilde{N}(1)\,\frac{\tilde{N}(1)}{\tilde{N}(2)} = \frac{\tilde{N}(1)^2}{N\left(\tilde{N}(1)\,/\,N\right)^2 e^{E_b/kT}} = g(1)\,N\,e^{-E_b/kT} \qquad \text{(II.6)}$$

The factor of 2 in g(1) accounts for the enhanced relative diffusivity associated with monomer / monomer impingement. N is the number of moles of atomic sites per unit volume. The binding energy, E_b, is here taken to represent the formation energy of a product phase dimer, not the nearest neighbor solute atom binding energy. E_b for Al_3Sc was calculated to be 0.129 eV using embedded atom method interatomic potentials described in [16].

Particles with radii greater than $r^* + \delta$ were considered to be thermodynamically stable and to evolve by diffusional growth. Expressions for g(i) and e(i) were derived by combining a mean-field, quasi-stationary approximation to the diffusion equation with the principle of detailed balance. (Related approaches were taken in the cluster dynamics model of Kampmann et al. [17] and in the work of Bales and Chrzan [9].) For the precipitates which have passed into growth, detailed balance is applied under the condition of global equilibrium; i.e. the point at which no further volume free energy change (corrected for strain) is available to drive the phase transformation. Thus, the constrained equilibrium quantities in Eq. II.3 are replaced by global equilibrium densities, $\Delta G_v + \Delta G_s$ in Eq. II.4 becomes zero, and Eq. II.4 reduces to

$$\frac{e(i+1)}{g(i)} = N^e(1)\,\exp\left[\frac{\gamma\,(A_{i+1} - A_i\,)}{k\,T}\right] \qquad \text{(II.7)}$$

The detailed balance expression is then combined with the solute flux to a size i particle (the balance of solute crossing the spatial shell designated by the size i+1 interface). Eq. II.8 sets the probabilistic flux, written in terms of the rate coefficients, equal to the deterministic flux.

$$g(i)\,N^0 - e(i+1) = D\,\nabla\,c(i+1)\cdot\hat{n}_{i+1} \qquad \text{(II.8)}$$

N^0, the monomer concentration available to attach to stable precipitates, was taken to include not only monomers, but also all solute tied up in embryos of size less than $r^* + \delta$, similarly to Kampmann and coworkers' approach [17]. N^0 is, therefore, representative of the instantaneous solid solution concentration. The deterministic flux is dependent on the gradient of the concentration in the vicinity of the precipitate/matrix interface, c(i+1), in the direction normal to the interface, n_{i+1}. c(i+1) is defined by the solution to the time independent diffusion equation, solved over a time interval during which the concentration field in the vicinity of the interface is assumed to be constant. The boundary conditions on the diffusion equation are set such that far from a given precipitate, the concentration is equal to N^0, while at the precipitate/matrix interface, the concentration boundary condition is set by the specification of an isotropic solute flux. The resultant deterministic flux for a spherical precipitate is

$$4\,\pi\,r_{i+1}\,D\left(\,N^0 - N^e_{i+1}\,(1,j)\right) \qquad \text{(II.9)}$$

where r_{i+1} is the sphere radius for a size i+1 precipitate. $N^e_{i+1}(1)$ is calculated as shown by Trivedi [18], using the common tangent construction between the solid solution free energy curve and the product phase free energy curve adjusted for the interfacial area and strain energy barriers. g(i,j) can now be written for stable precipitates as (with e(i,j) defined from Eq. II.7)

$$g(i,j) = 4\,\pi\,r_{i+1}\,D\,\frac{N^0\ - N^e_{i+1}\,(1,j)}{N^0\ - N^e\,(1)\,\exp\,\left[\,\gamma\,(A_{i+1} - A_i\,)\,/\,kT\right]} \qquad \text{(II.10)}$$

Finally, the impact of thermally generated defects on the diffusivity is discussed. Sundar and Hoyt [19] addressed this issue by defining a time dependent solute diffusivity arising from an estimated vacancy decay rate due to a fixed vacancy sink density (dislocation density, ρ_d). The same effect was treated in the present work by an additional rate equation, taken from Adams and Wolfer [9], which was solved simultaneously with the solute rate equations.

$$\frac{d\,v(1)}{d\,t} = -2\,\pi\,\rho_d\left[\ln\left(1/2\,a\,\sqrt{\pi\,\rho_d}\,\right)\right]^{-1} D_v\left(\,v(1) - v^e(1)\,\right) \qquad \text{(II.11)}$$

The vacancy diffusivity, D_v, and the thermal equilibrium value of the monovacancy concentration, $v^e(1)$, are temperature dependent and written in the standard fashion as functions of the vacancy formation energy (~0.67 eV for Al) and the vacancy migration energy barrier, equal to the experimentally determined self-diffusion activation energy (~1.28 eV for Al) minus the vacancy formation energy [20].

Simulation Results

The Al:Al_3Sc system was chosen for the preliminary test of the reaction rate model for several reasons. First, both diffusion data [21] and thermodynamic data [22] are available for Al-Sc alloys and recent experimental measurements have been made of the kinetics of nucleation of Al_3Sc [5] for comparison with the present calculations. Second, the Al_3Sc phase forms fully coherent interphase boundaries with the matrix, making the assumption of isotropic atomic attachment a reasonable one.

Figure 2 compares TEM observations of nucleation of Al_3Sc in an Al-.11 at% Sc alloy with the simulated "observable" (diameter > 10Å) Al_3Sc number densities during a quench from 923K (above the Al_3Sc solvus) to a range of isothermal reaction temperatures. The reaction temperatures

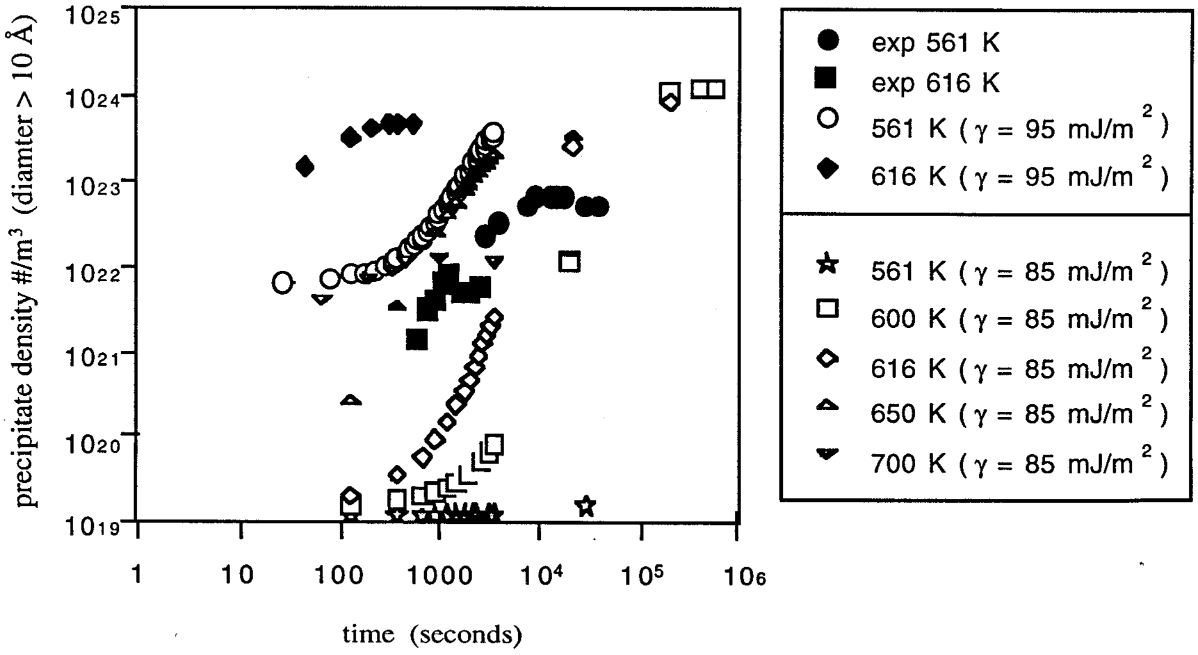


Figure 2. Early Al_3Sc evolution in Al-0.11 at. % Sc.

were reached by 1 sec, typical of the thin samples used in the TEM study. Simulation results shown by open symbols were obtained using $\gamma = 85$ mJ/m^2 and $E_a = 2.32$ eV. These energies were not optimized to match the experimental data, as evidenced by too high number densities and too slow kinetics.

To show the impact of varying the input data, γ was increased to 95 mJ/m^2 and E_a was decreased to 1.99 eV (as suggested in [5]). The results, represented by the dotted symbols, do not reproduce the experimental data, but do decrease the number densities and increase the kinetics, as expected. Overall, the temperature dependent trends agree with experiment. The maximum in Al_3Sc density is attained faster at the higher temperatures, consistent with the higher solute diffusivities, and the magnitude of this maximum is larger with lower temperatures, consistent with the larger supersaturation driving the phase transformation.

The precipitate size distribution as a function of reaction temperature is shown in Figure 3. The average size of the precipitates decreases with decreasing temperature or increasing supersaturation, as expected. The influence of size coarsening at the higher temperatures is demonstrated by the increased number densities of larger precipitates at the expense of the smaller precipitates.

The impact of excess vacancies on the precipitation process is indicated in Figure 4 which shows the vacancy supersaturation, $C_v - C_v^{eq}$, as a percentage of the thermal equilibrium vacancy content, C_v^{eq}, together with the evolution of the "observable" Al_3Sc density for two vacancy sink densities. The enhanced vacancy content resulting from the lower sink density increases the precipitation kinetics, but does not change the final number density of precipitates observed. The Al_3Sc density evolution for this case can be explained as follows. The rapid increase in density up to 500 seconds is due to the high vacancy content and associated high diffusivity up to that time which

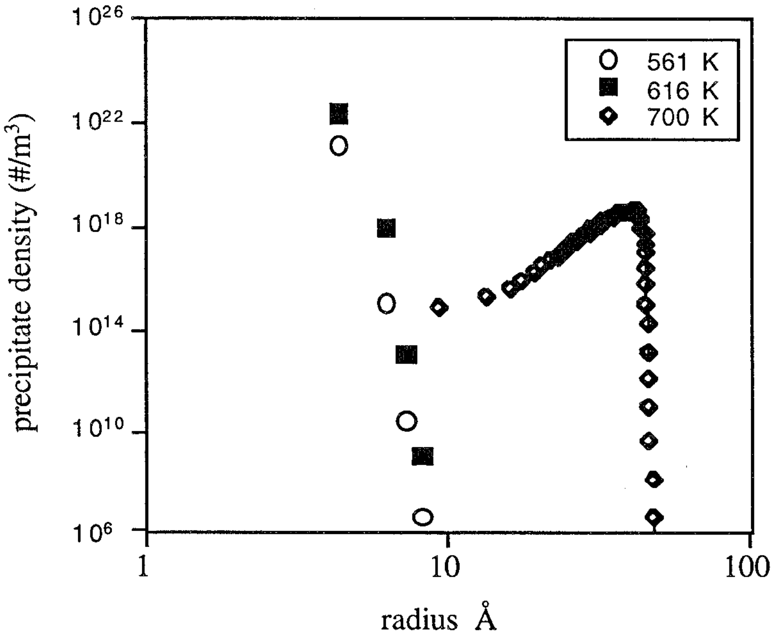


Figure 3. Simulated Al_3Sc size distributions after one hour at varied reaction temperatures.

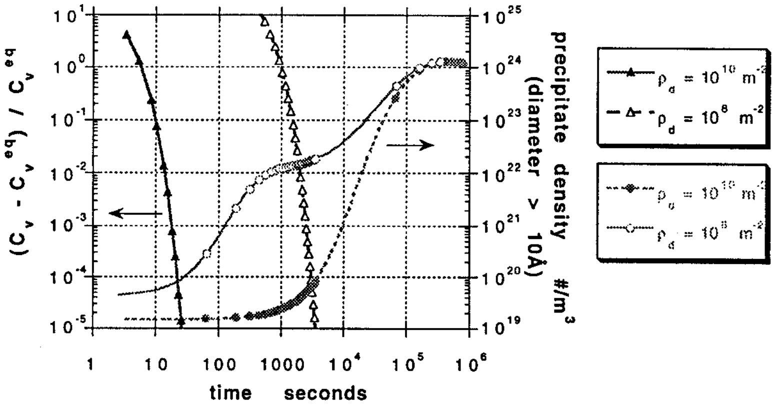


Figure 4. The influence of decreased vacancy sink density on precipitation in Al-0.11 at. % Sc.

greatly enhance the nucleation and growth rates. A decrease in the kinetics is then observed as the vacancy content begins to drop off. The next density increase occurs as new precipitates nucleated at the now slower rate grow past the 10 Å cut-off. By 10^5 seconds, the solute supersaturation (not shown) has begun to drop significantly, the production rate of new nuclei has dropped, and the number density of larger particles flattens out. The evolution for the high sink density case reflects the attainment of a nearly time invariant diffusivity due to the much earlier drop in vacancy content.

Conclusions

A kinetic model of thermal history dependent precipitation based on reaction rate theory has been presented and applied to precipitation in an Al-Sc alloy. The model was shown to represent the processes of nucleation, growth, and size coarsening in a single framework using generalizable alloy thermodynamics. Simulation results were shown for varied thermodynamic energies (γ, E_a), reaction temperatures, and vacancy supersaturations, indicating the flexibility of the approach in studying the possible impact of varied thermal processing paths and assumed atomic assembly mechanisms on the transformation kinetics.

III. Embedded-Atom-Method Study of Al:Al_3Sc Interphase Boundaries

Although many of the theoretical aspects of homogeneous nucleation in solids are fairly well established [23-26], homogeneous nucleation is a relatively rare phase transformation in the majority of metallic alloy systems of commercial interest. However, the metallurgical benefits obtained from Al alloys containing Sc additions in the range of 0.2 at% [27-32] have been shown by direct microstructural analysis to result from the reaction fcc→$L1_2$ in which fully coherent, ordered crystals of Al_3Sc form homogeneously from the matrix [3]. In view of the potential commercial importance of this reaction and its relatively simple thermodynamic and crystallographic aspects, it was judged to be desirable to use the fcc→$L1_2$ transformation in Al rich Al-Sc as a model system for investigating the interfacial structure and chemistry associated with the formation of a strongly ordered intermetallic phase in an aluminum matrix.

The work presented here is a theoretical continuation of previous experimental work by Hyland [3] on the homogeneous nucleation kinetics of $Al_3Sc(L1_2)$ in a dilute Al-Sc alloy. This work was based on TEM observations of the number density of Al_3Sc precipitates versus isothermal reaction time at 561 K and 616 K that permitted the measurement of nucleation rates. Using classical nucleation theory, an average Al(fcc):$Al_3Sc(L1_2)$ interphase boundary energy, γ, was subsequently back-calculated from the measured nucleation rate data. This average γ was compared with separate calculations based on classical thermodynamic theory for three low index interface orientations, using the Becker model [33] for the chemical component of γ.

A number of assumptions attended the determination of both the average γ extracted from the experimental data and the calculations of γ for individual boundary orientations. First, the nuclei were assumed to be classical in nature and, therefore, to possess physical properties representative of a uniform second phase whose total free energy can be safely partitioned into volume and surface contributions. Second, the fcc solid solution was assumed to follow simple regular solution thermodynamics, and the precipitate phase was assumed to be stoichiometric at 25 at% Sc.

Additional assumptions specific to the Al-Sc system were made in both approaches. The average γ values calculated from experimental nucleation rate data were based on the assumption that the anisotropy of the interfacial energy was negligible in the temperature range where the experiments were conducted. Under conditions where classical theory applies, it is usually the anisotropy of γ that is of overriding importance in determining the shape of the critical nucleus and, therefore, the overall barrier height to nucleation. Hence, the assumption of an isotropic interphase boundary energy requires some justification with increasing undercooling.

The calculations based on the Becker model also involved several additional assumptions. First, the composition gradient normal to the fcc:$L1_2$ interface was assumed to be a step function, and the compositions of each bulk phase were, therefore, assumed to be uniform throughout. This approximation neglects contributions to the total free energy from gradient energy terms [23, 34] that are expected to lower the interfacial energy at finite temperatures. Second, each bulk phase is assumed to be disordered in the Becker model; the neglect of long-range order in the Al_3Sc phase is also expected to lead to an overestimate of γ. Third, the atomic interaction energies were derived from continuum level bonding enthalpies for Al and Sc. Finally, the temperature dependence of γ was assumed to be relatively weak based upon the Al-Sc phase diagram [22] which shows very little variation in the phase boundary compositions with temperature. This is an indication that the chemical component of γ should be only weakly temperature dependent and, in view of the highly ordered state of the $L1_2$ phase [35], entropy contributions are expected to be small.

Classical nucleation theory is based on the assumption that a critical nucleus is large enough that its free energy of formation can be partitioned into a surface and a volume free energy contribution. The work of LeGoues et al. [24-25] has shown that classical theory is applicable to critical nuclei of Co rich precipitates in dilute Cu-Co alloys reacted at low to moderate (i.e. less than 200 K) undercooling. A reasonable criterion for the applicability of classical nucleation theory, first suggested in [36] and later modified in [25], is that the dimensions of a critical nucleus must be greater than the thickness of a compositionally diffuse interphase interface at a given reaction temperature. As noted by LeGoues, Lee and Aaronson [25], a classical nucleus is expected to be one that attains a constant composition at some point throughout its volume, though their work suggested that it may not be necessary for the entire nucleus to be compositionally uniform in order for classical nucleation theory to be applicable. If the nucleus composition varies continuously throughout its volume, the use of the classical approximation is inappropriate for establishing the energetics of critical nuclei and for comparing nucleation theory with experiment.

It is known that as the relative undercooling below the solvus is increased the critical nucleus dimensions decrease, making the application of classical models of critical nuclei problematic. At large enough undercoolings, the nuclei become small enough that they are probably not compositionally uniform at any point. In such cases, the critical nuclei are considered to be nonclassical, and analysis of experimental nucleation rate data requires the use of a theory that can treat nonuniform phases. The continuum nonclassical theory of nucleation due to Cahn and Hilliard [36] is well developed for cases where the free energy of the system can be written as a continuous function of the composition and its spatial derivatives. In the Al-Sc system, such an approach is of limited utility since the Al_3Sc phase is highly stoichiometric, while the fcc solid solution exhibits a vanishing solubility of Sc. These features of the Al-Sc system imply that the interphase boundaries will be fairly sharp compositionally on an atomic scale, thus making difficult a continuum nonclassical formulation of the two-phase microstructure. For this reason, an alternate means of evaluating the composition profile normal to an fcc:$L1_2$ interphase boundary is desirable in order to determine whether nuclei are forming classically or nonclassically at a given temperature.

While the spatial extent of a compositionally diffuse interface cannot be probed readily via experiment, it can be estimated by appealing to calculations based on gradient thermodynamics [23-26], or via atomistic simulation of the interfacial region, if the interatomic potentials can be adequately described. In view of the availability of many-body, volume-dependent interatomic potential models such as the embedded atom method (EAM) [4], atomistic simulations of many metallic systems are desirable. Furthermore, atomistic simulations obviate all of the classical assumptions listed above. In the present work, EAM-based, atomistic simulations and a low-temperature expansion (LTE) approach, to be described below, were used to determine the applicability of classical versus nonclassical nucleation theory in interpreting the available experimental data in dilute Al-Sc alloys. The LTE method was used to calculate the temperature dependence of the $(100)_{fcc} \| (100)_{L12}$ interphase boundary energy. Atomistic energy minimization at 0 K and Monte Carlo simulations at 573 K (chosen to coincide roughly with the temperature range used in the experimental work [3]) were used, respectively, to study the orientation dependence of the excess enthalpy and to calculate the width of the probable compositional diffuseness present at the interphase boundaries. A comparison of the predicted width of the compositional gradient to the critical nucleus size calculated from the experimental nucleation rate data should allow for an assessment of the applicability of classical versus nonclassical nucleation theory to the Al_3Sc ($L1_2$) phase in the Al-Sc binary alloy in the temperature range used.

In the next subsection, details of the interatomic potentials used in this study are presented along with a discussion of the ability of these potentials to reproduce established thermodynamic behavior. The subsequent two subsections discuss the low and high temperature calculations, respectively. Finally, the likelihood of observing classical nucleation in the temperature range studied in the experimental measurements [3] of the nucleation rate of the $Al_3Sc(L1_2)$ phase in dilute Al-Sc alloys is discussed.

Details of the Interatomic Potentials

A rough set of EAM [4] interatomic potentials were developed for the Al(fcc):$Al_3Sc(L1_2)$ interfacial energy calculations. The potentials were optimized neither for pure Al nor for pure Sc. The Al potential was taken from Voter and Chen [37]. For simplicity, the Sc potential was fit assuming an equilibrium fcc crystal structure, rather than its standard state hcp structure. This assumption seems reasonable because the properties of the fcc solid solution and of the $Al_3Sc(L1_2)$ phase were the main focus of the potential development, both of which incorporate Sc in an fcc environment.

Calculated properties of Al, Sc and Al_3Sc are compared to experimental information in Table I. In

TABLE I. CALCULATED VS. EXPERIMENTAL MATERIAL PROPERTIES

	Al		Sc		Al_3Sc	
	calculated	experiment	calculated	experiment	calculated	experiment
B (10^{11} Pa)	0.793	0.793	0.572	0.572	1.011	0.992
A	1.529	1.213	3.760	1.000		
$<G>_v$ (10^{11} Pa)	0.277	0.294	0.284	0.298		
C_{11} (10^{11} Pa)	1.073	1.14	0.714	0.969		
C_{12} (10^{11} Pa)	0.653	0.619	0.501	0.373		
C_{44} (10^{11} Pa)	0.321	0.316	0.401	0.298		
E_{vac} (eV)	0.637	0.67	1.442	1.4		
$\Delta E_{bcc-fcc}$ (eV)	0.073	0.10	0.015			
$\Delta E_{hcp-fcc}$ (eV)	0.014	0.06	-0.003			
H (eV)			-0.663	-0.62		
E_{sub} (eV)					-0.259	-0.4
Lat. Param. (Å)					4.101	4.106
SISF (erg/cm^2)					256.3	265
(100) APB (erg/cm^2)					330.2	450
(111) APB (erg/cm^2)					433.9	670

the table, B, A and <G>$_v$ are the bulk modulus, the anisotropy ratio and the Voigt averaged shear modulus, respectively, derived from the elastic constants as follows:

$$B = \frac{C_{11} + 2\,C_{12}}{3} \qquad \text{(III.1)}$$

$$A = \frac{2\,C_{44}}{C_{11} - C_{12}} \qquad \text{(III.2)}$$

$$\langle G \rangle_V = \frac{3\,C_{44} + C_{11} - C_{12}}{5} \qquad \text{(III.3)}$$

A reasonable, if not ideal, fit between calculated and experimental properties [22, 39-41] was obtained. Note that the calculated value of the pure Sc fcc to hcp transformation energy, $\Delta E_{hcp\text{-}fcc}$, is negative. This is an interesting result because even though the potentials were fit assuming that Sc atoms occupy ideal fcc sites, the calculated value of $\Delta E_{hcp\text{-}fcc}$ indicates that the hcp structure should be more stable than the fcc structure for Sc, in agreement with experiment at low temperatures.

A series of Monte Carlo (MC) calculations were carried out to study the stability of the $L1_2$ phase and to test the ability of the Al-Sc potentials to predict the appropriate equilibrium phases at a specified combination of temperature, pressure and chemical potential difference, $\Delta\mu = (\mu_{Sc} - \mu_{Al})$. This test was also applied to potentials for Al, Ni and Ni_3Al by Foiles and Daw [42]. The MC calculations were carried out assuming the system is always in thermodynamic equilibrium. Therefore, in two phase regions, the chemical potentials of the individual atomic species i, in each phase, $\mu_{i,a(b)}$, must be equal by definition; i.e., $\mu_{i,a} = \mu_{i,b} = \mu_i$. Also by definition, the chemical potentials of the individual species must be constant as a function of composition across a two phase region, thus making $\Delta\mu$ constant across this region as well. Al and Sc form a series of line compounds separated by two phase regions [22]. Thus, a plot of $\Delta\mu$ versus Sc content should show a series of horizontal lines separated by sharp vertical transitions at the boundaries between two phase and single phase regions. Our goals were to test whether the fitted potentials could generate such a plot and to identify the appropriate value of $\Delta\mu$ for the two phase Al(fcc):$Al_3Sc(L1_2)$ region of interest in this study.

The procedure involved setting up a two phase "box" containing a total of 864 atoms, Al(fcc) in one-half of the box and $Al_3Sc(L1_2)$ in the other, finding the 0 K minimum energy atomic configuration using a conjugate gradient descent technique [43], and then conducting a series of MC simulations at 573 K, varying $\Delta\mu$. The total number of atoms was held constant, but the relative number of Al atoms to Sc atoms was allowed to change in accordance with the requirements of an open system using the specified values of $\Delta\mu$. A MC step comprised either a) a random spatial adjustment of a randomly chosen atom plus a replacement of its atomic species (Al->Sc or Sc->Al) or b) a random adjustment of each of the three box lengths. Step type b) allows for a change in total volume, which is effectively comparable to the condition of zero applied pressure. Each simulation was equilibrated for 2,500,000 steps, and the composition of the system was averaged over a subsequent 1,000,000 steps. Periodic boundary conditions were enforced on the computational box, the effects of which are discussed in the next section.

The black dots in Figure 5 indicate the resulting averaged composition as a function of $\Delta\mu$. Note that the axes have been reversed, making $\Delta\mu$ the ordinate rather than the abscissa, in order to more clearly match the expected $\Delta\mu$ versus at% Sc diagram described above. As shown in the figure, large ranges in $\Delta\mu$ resulted in single-valued Al:Sc compositions. These compositions correspond to the compositions of the line compounds, the boundaries between two phase regions. The

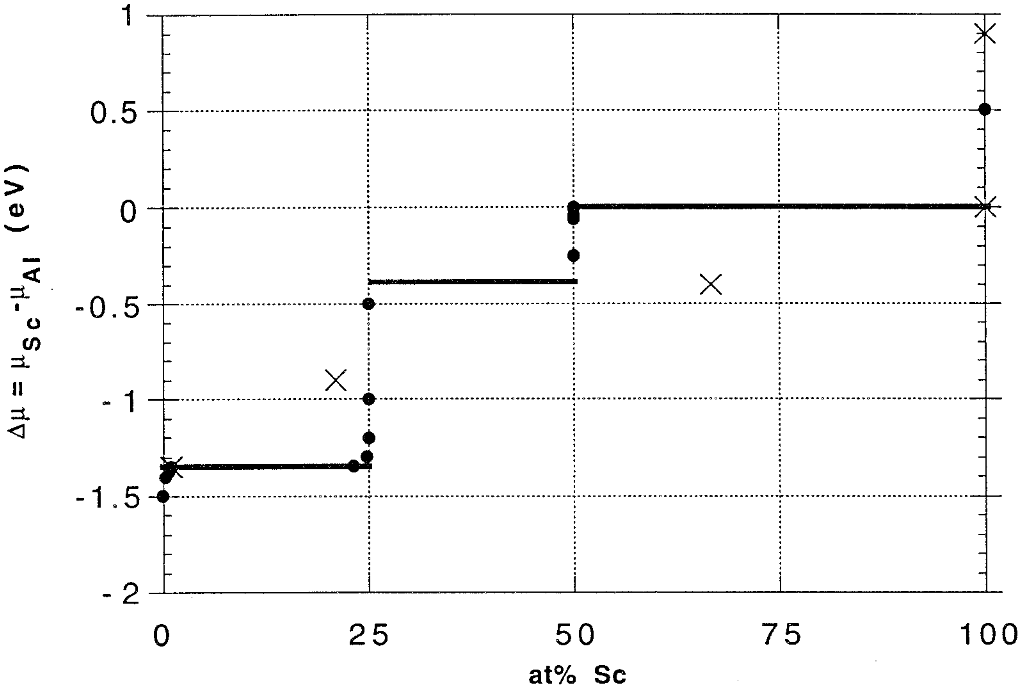


Figure 5. Difference in chemical potentials, $\Delta\mu$, versus atomic % Sc.

potentials did generate both the compositions and crystallographic structures of the Al(fcc), $Al_3Sc(L1_2)$, AlSc(B2) and Sc(fcc) phases. The Al_2Sc and $AlSc_2$ phases, however, never materialized. This was not surprising, as the interatomic potentials were not fit for either of these two phases. Furthermore, the constraint on the simulation box volume allowing the lengths of the box axes to change, but not the angles between axes, should have hindered the formation of the hcp $AlSc_2$ phase. The Al_2Sc phase, on the other hand, is cubic, but possesses a 24 atom unit cell. The formation of such a complex phase is probably beyond the capabilities of the present simulations. Nevertheless, to assure that the two phase initial atomic configuration used in the simulations was not biasing the equilibrated state, additional simulations were run starting with a random fcc solid solution of 61 at% Sc. The results from this set of simulations are designated in Figure 5 by crosses. All of the final compositions are in good agreement with the original set of simulations except for the composition calculated for $\Delta\mu$ equal to -0.4. The resulting mole fraction of Sc would lead one to believe that the $AlSc_2$ phase had been created, but, in fact, the resulting atomic configuration was not single phase. Instead, large blocks of pure fcc Sc were separated by layers of Al. This result illustrates the sensitivity of the equilibrated state obtained from the chosen interatomic potentials to the initial atomic configuration. Nonetheless, the remaining results were consistent, and for the purposes of the present work, the chemical potential difference for the $fcc{:}L1_2$ two phase region of interest was established at $\Delta\mu \sim -1.3$ eV.

Little emphasis is placed upon differences among the individual absolute values of the calculated interfacial energies to be discussed in the following section because of the nonideal fit of the potentials to the limited experimental material properties and thermodynamic data. We expect, however, that the reasonable fit to the available information demonstrated above should result in a valid representation of the qualitative behavior of the interphase boundary energies thus calculated.

Low Temperature Interphase Boundary Energy Calculations

The total energy of a system containing a coherent interphase interface separating two phases that posess different molar volumes can be written as a sum of the contributions from a) the total energies of the two isolated phases, b), the chemical work associated with the creation of the composition gradient between phases and c), the energy associated with atomic and interplanar displacements at the interface. The difference between the sum of the last two factors and the volume-dependent elastic strain energy is the classically defined interphase boundary energy, γ. An advantage of atomistic simulation is its ability to treat the chemical and atomic relaxation contributions to γ consistently without making the assumptions required in continuum level treatments.

The energies of systems containing (100), (110) and (111) Al:Al_3Sc interfaces were minimized at 0 K. Because the unconstrained lattice parameters of Al and Al_3Sc differ by approximately 1.4%, being 4.05 Å and 4.105 Å for Al and Al_3Sc, respectively, the bulk phase reservoirs are expected to deviate from their far field, undistorted, molar volumes in the vicinity of the interface. Simulations of the chosen interfaces were set up with total number of atoms and total number of bulk atomic layers parallel with and on either side of the interface as listed in Table II. The initial configuration was established with a uniform lattice parameter of 4.05 Å in both phases and a perfect match at the interface. Periodic boundary conditions in all directions were employed, making the interface effectively infinite in extent, but producing "image" interfaces parallel to the actual interface. The periodic boundary conditions also caused the outer face of the simulation box parallel to the actual interface to be a second explicit interface. Thus, all calculated excess energies are divided by two and are representative of the average energy of the two explicit interfaces. Figure 6 is a schematic of the simulation conditions.

It is important to emphasize the simulation conditions for the following reason. Based on Figure 6, it is clear that atomic relaxation in the direction perpendicular to the interface is not artificially constrained by the imposition of periodic boundary conditions. However, the box faces perpendicular to the plane of the interface are constrained to remain flat as a consequence of these boundary conditions. In other words, far from the interface, the lattice constants of the two reference phases parallel to the interface are artificially required to be identical. They are inhibited from fully relaxing to their equilibrated values of 4.05 Å and 4.105 Å. Therefore, neither reservoir attains its undistorted unit cell dimensions far from the interface. This enforced condition is expected to raise the total energy of a system containing an interface by some amount over the value that would be obtained if 3 dimensional relaxation was permitted.

In order to calculate a barrier height for diffusional nucleation we require the chemical component of the interphase boundary energy. To ascertain this quantity, which is independent of the size of the computational system under study, the following procedure was adopted. The energy of the system containing the interphase boundary was minimized by allowing atomic and volumetric relaxation at zero applied pressure, subject to the boundary conditions discussed above. Similar energy minimizations were carried out on the bulk Al and Al_3Sc phases set up with the same orientations as the interface calculations and using the same total number of atoms. In the simulations of the bulk Al and Al_3Sc, the lattice constants parallel to the interphase boundary were fixed at the value obtained from the system containing the interface. As a consequence of the imposed boundary conditions, the energies of the elastically distorted bulk reservoirs were calculated by permitting relaxation normal to the interface. The interfacial energies, γ, were subsequently calculated as

TABLE II. DETAILS OF INTERFACE SIMULATIONS

	(100)	(110)	(111)
total # of atoms:	2560	3584	3024
# Al:	2240	3136	2646
# Sc:	320	448	378
# of layers in each phase parallel with and on either side of the interface:	10	16	9

$$\gamma = \frac{E_{ipb} - \frac{1}{2}(E_{Al} + E_{Al_3Sc})}{2A} \qquad \text{(III.4)}$$

with the values of E_i's being the energies of the interface containing systems and of the two elastically distorted bulk reference phases, respectively, and A being the interfacial area. The 0 K excess energies for each of the three crystallographic interfaces at zero applied pressure are listed in Table III. As shown in the table, $\gamma_{(100)} < \gamma_{(110)} < \gamma_{(111)}$, with $\gamma(110) > (2)^{1/2}\,\gamma(100)$ and $\gamma(111) > (3)^{1/2}\,\gamma(100)$. The ratios of these calculated interfacial energies dictate that both the (111) and (110) orientations are unstable with respect to (100) faceting at 0 K. The calculated results from [5] are presented in Table III for comparison. Note that the trend in energies is opposite to that calculated atomistically. The Becker model predicts that (111) facets should be significantly more stable than (100) or (110) facets. If the Becker model predictions were correct, one would expect $(111)_{L12} \parallel (111)_{fcc}$, reflecting the low energy of the (111) interface, since orientation relationships are presumably set at nucleation and survive into growth. Experimentally, this orientation relationship is not observed. Repeated TEM investigations [5, 30-32] have shown that Al_3Sc precipitates nucleate with a cube:cube orientation relationship and do not demonstrate an orientation relationship defined by (111) conjugate habit planes. The atomistic calculations described in the present work are more consistent with these experimental observations.

Also listed in Table III are the calculated lattice parameters both perpendicular and parallel to the interface. The perpendicular lattice parameters are seen to be close to the average between the pure

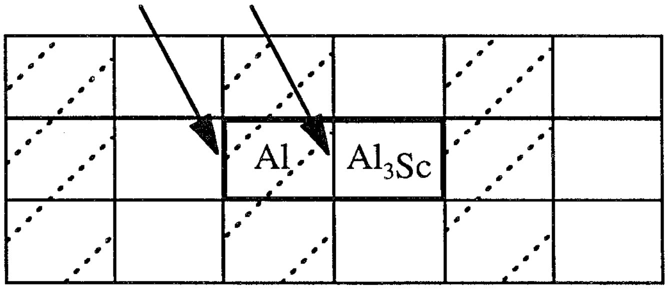


Figure 6. Schematic depiction of interface simulation set-up with periodic boundary conditions.

TABLE III. CALCULATED AND PREVIOUSLY ESTIMATED [5] VALUES OF Al/Al_3Sc INTERPHASE BOUNDARY ENERGIES.

	(100)	(110)	(111)
γ (mJ/m^2)	32.5	51.3	78.2
(from [5])	(126)	(112)	(28)
$a_{\parallel}$	4.080	4.086, 4.077	4.084
$a_{\perp}$	4.071	4.069	4.066

Al and pure Al_3Sc lattice parameters, as expected, since the bulk phase lattice constants are permitted full relaxation perpendicular to the interface. The values of the lattice parameters parallel to the interface, however, are those that result from the minimization of the total energy subject to the periodic boundary conditions.

High Temperature Interphase Boundary Energy Calculations

To examine the Sc composition profiles near the interphase interface, MC simulations were carried out at 573 K for each of the three orientations, and low temperature expansion (LTE) calculations were performed as a function of temperature for the (100) interface. The MC simulations were carried out with zero applied pressure, constant temperature and constant number of atoms. Unlike in the MC simulations described earlier, atomic MC steps involved randomly choosing *two* atoms, giving each a random spatial adjustment and then *exchanging* their respective element types. This kind of step mimics long-range, volume interdiffusion, without regard for the path an atom may have taken, but holds constant the relative number of Al atoms with respect to the number of Sc atoms (closed system). The initial atomic configurations used were the 0 K, minimum energy interface and bulk configurations described in the previous section. Each configuration was subsequently equilibrated at 573 K for 5,000,000 MC steps, followed by a 5,000,000 step period over which the total enthalpy, volume, parallel and perpendicular lattice parameters, and atomic positions were averaged.

To determine the temperature dependence of the total interfacial energy, including entropic contributions, LTE calculations of finite-temperature grand potentials (Ω) [44] were performed for the bulk Al and Al_3Sc phases, as well as for the (100) Al:Al_3Sc interphase boundary. In the LTE method [45], the grand potential is calculated directly from a Taylor series expansion of the logarithm of the alloy partition function. To second-order, the LTE expression for the grand-potential has the following form:

$$\begin{aligned}\Omega = E_0 &- k_B T \sum_{p} exp\left(-\Delta\omega_p \;/\; k_B T\right) + 1/2\; k_B T \sum_{p} exp\left(-2\;\Delta\omega_p \;/\; k_B T\right) \\ &- 1/2\; k_B T \sum_{p,p'} \left\{exp\left(-\Delta\omega_{p,p'} \;/\; k_B T\right) - exp\left(-\left[\Delta\omega_p + \Delta\omega_{p'}\right] / \; k_B T\right)\right\}\end{aligned} \qquad \text{(III.5)}$$

where the sums are over lattice sites p, and where k_B and T represent Boltzmann's constant and the temperature, respectively. In Eq. (III.5), E_0 represents the 0 K grand potential, and the variables

$\Delta\omega_p$ and $\Delta\omega_{p,p'}$ represent excitation energies. $\Delta\omega_p$ denotes the change in the zero-temperature grand-potential associated with switching the atom type at site p. Similarly, $\Delta\omega_{p,p'}$ is the cost in the zero-temperature grand potential associated with changing atom types at both sites p and p' (note that for close-neighboring sites, p and p', $\Delta\omega_{p,p'} \neq \Delta\omega_p + \Delta\omega_{p'}$).

Formally, the finite-temperature interfacial energy for an (hkl) interface can be written as follows [44]: $\gamma_{hkl}=(\Omega_{hkl} - \Omega_0)/A$, where Ω_{hkl} is the value of the grand potential for an inhomogeneous alloy system containing an (hkl)-oriented interphase boundary, and Ω_0 corresponds to the elastically distorted homogeneous bulk phase reservoirs. Using the LTE approach, the grand potential difference can be determined once the excitation energies have been computed for all symmetry-inequivalent points in the bulk phases, and for all points inside a region near the boundary where the excitation energies have values which differ from those in either of the corresponding bulk phases.

In the current calculations, we have included terms in the low-temperature expansion associated with all single-atom excitations and two-atom substitutions for pairs of atoms separated by a distance less than or equal to the fourth neighbor shell. The inclusion of these terms was found to lead to an expansion for the grand potential difference which was reasonably well converged (the contribution from the second-order term is less than 1/3 that of the first-order term contribution) for temperatures up to 800 K. Excitation energies were calculated directly from the EAM potentials described earlier using a supercell geometry. The supercell included a total of 12 unit cells normal to the interphase boundary and 6 unit cells in each of the in-boundary plane directions, requiring a total of 1728 atoms. The atomic positions were relaxed to minimize the energy after each atomic rearrangement. Note that in calculating the values of the excitation energies, temperature effects associated with atomic vibrations were neglected. As a consequence, the LTE calculations of the alloy grand potentials include configurational, but not vibrational entropy contributions.

In Figure 7, the total interfacial energy, γ, enthalpy, H, and entropy contribution, -TS, are plotted as a function of temperature for the (100) interphase boundary. It is seen that the excess free energy is practically temperature independent up to approximately 400 K, and then decreases by approximately 10% from 400 K to 800 K. The decrease above 400 K can be attributed to the significant configurational entropy contribution at the higher temperatures which accompanies increasing compositional diffuseness at the interface. The experimentally determined average γ is 93±22 mJ/m^2 at 616 K and 78±20 mJ/m^2 at 561 K [3], compared to the LTE values of 39.1 mJ/m^2 at 616 K and 39.7 mJ/m^2 at 561 K , respectively, for the (100) interface. In spite of the nonideal fit of the interatomic potentials, the calculated values of $\gamma_{(100)}$ seem reasonable with respect to the experimental results that were secured from the application of classical nucleation theory to the nucleation rate data.

An important influence on the high temperature interfacial thermodynamic properties is the composition gradient across the interfaces. The average compositions of atomic layers parallel to the interfaces were determined from the MC results and are graphically represented in Figure 8. For each orientation, distinctly bulk-like regions were found on either side of the interfaces separated by approximately four atomic layers of compositional diffuseness. The chemical diffuseness was most pronounced for the highest energy (111) interface followed by the (110) and the (100) interfaces.

Two issues are raised pertaining to the Sc distributions in the atomistic calculations and, therefore, to the calculated interphase boundary energies. The first issue is whether or not enough layers parallel to the interfaces were included in the calculations to actually have large enough bulk-like regions so that the explicit interfaces were both fully relaxed and non-interacting. The second is that, even though Sc solubility in Al is negligible, it is not zero. The MC simulations discussed thus far involved pure Al in contact with pure Al_3Sc. Therefore, the slight solubility of Sc on the Al side of the reservoir bounding the interfaces could only come from the bulk Al_3Sc, rather than

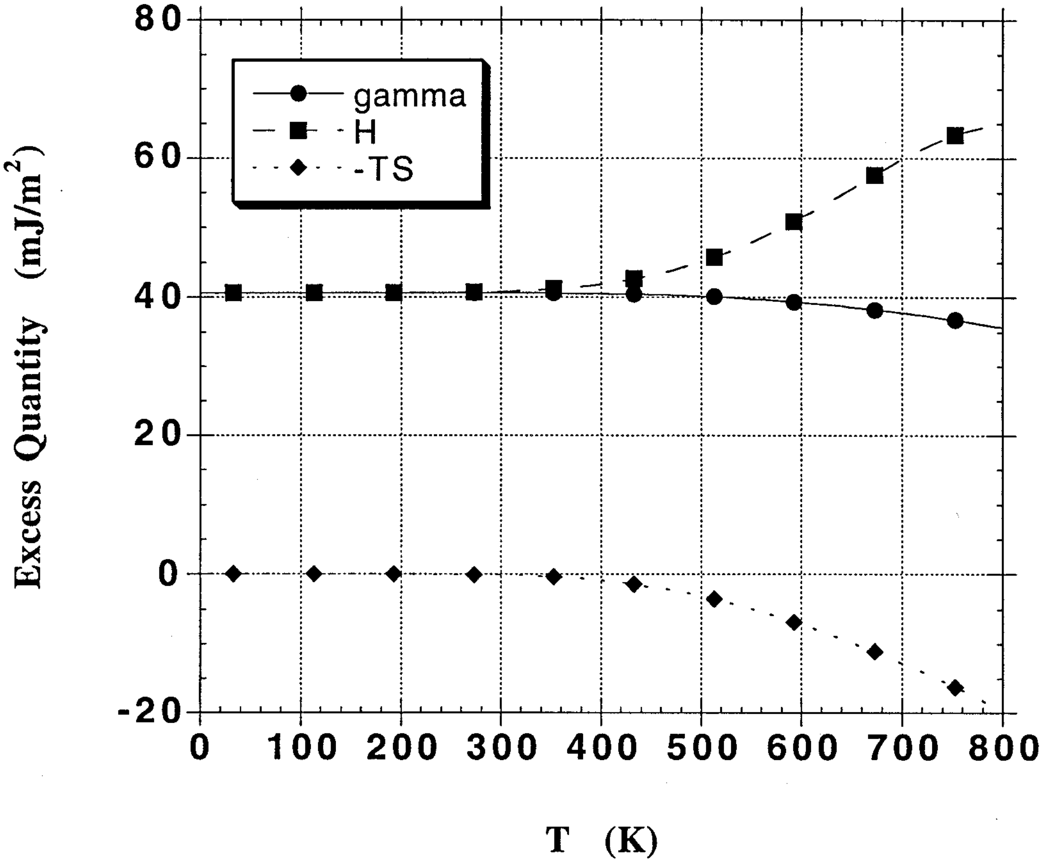


Figure 7. Excess thermodynamic properties of the {100} interphase boundary orientation in Al(fcc):$Al_3Sc(L1_2)$ based on the LTE method.

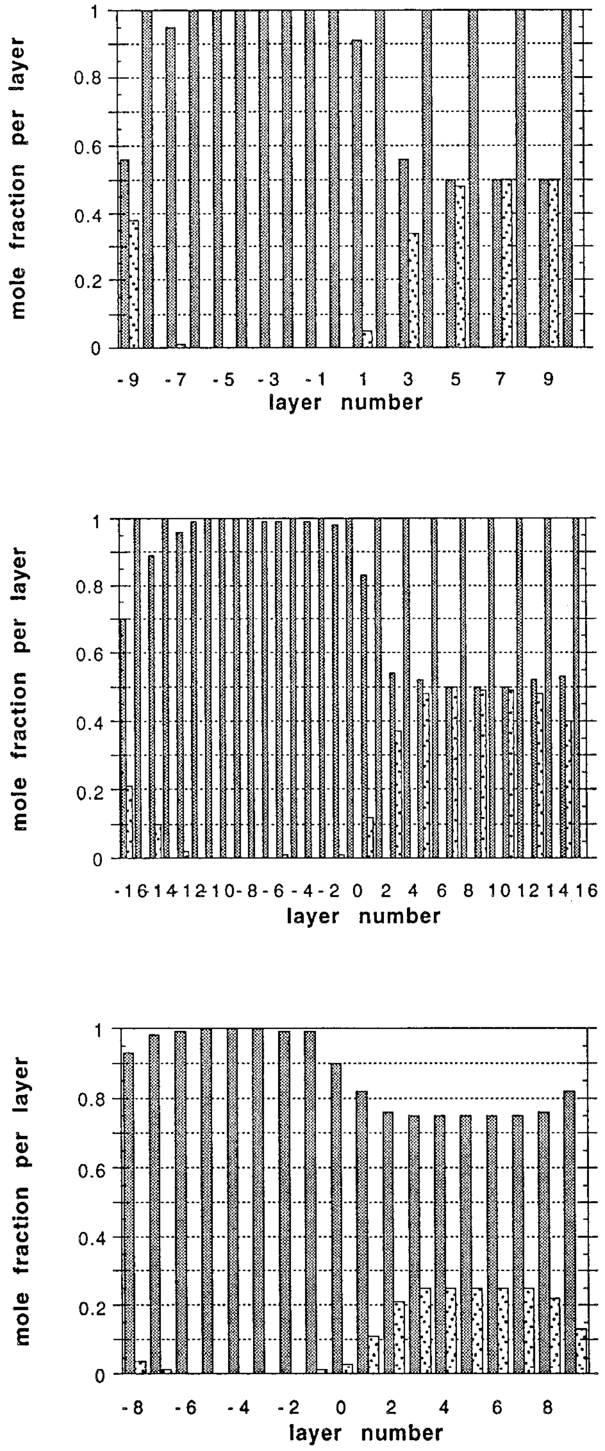


Figure 8. From top to bottom: (100), (110), and (111) interphase interface composition profiles at 573 K. Dark bars represent mole fraction of Al. Light bars represent mole fraction of Sc.

from equilibrium segregation of Sc in solid solution to the interface. Several calculations were carried out to address these two issues.

To address the first issue, a second set of 0 K energy minimizations and 573 K MC simulations of the Al:Al_3Sc (100) interface were carried out with 18 bulk atomic layers parallel to and on either side of the interface, as opposed to the 10 layers used in the previous (100) interface calculations. 4608 atoms were included in these calculations, as opposed to 2560. The interfacial area was the same as that of the previous system. The 0 K interfacial excess enthalpy was calculated to have the same value, 32.5 mJ/m^2, as was obtained from the smaller system. The MC calculated composition profile at 573 K for the larger system is shown in Figure 9. Again, two bulk-like regions are separated by four atomic layers of compositional diffuseness. From these results, it is concluded that the smaller simulation systems are capable of representing non-interacting interfaces separated by bulk phases of infinite extent.

To investigate the temperature dependence of the fcc bulk phase composition and its effect on the spatial extent of the compositionally diffuse region associated with the Al:Al_3Sc interphase boundaries, two 573 K MC simulations of the (111) interface orientation were carried out with 0.25 and 1.0 at% Sc in Al solid solution. Both of these compositions are beyond the solubility limit predicted by the phase diagram. Therefore, the majority of the excess Sc was expected to create additional Al_3Sc, the remainder being necessary to establish the appropriate chemical potential of Sc in the fcc phase in the limit of a nonvanishing solid solubility. For all simulations, the Sc was initially randomly distributed in the solid solution, the systems were energy minimized at 0 K, and then the MC procedure discussed above was carried out. Composition profiles from these two cases showed little difference from that of the pure Al:Al_3Sc interface as shown in Figure 10. Hence, it is reasonable to conclude that the calculated composition profiles are only weakly affected by the exact solubility of Sc in Al because a truly dilute solid solution exists, even at relatively high values of the homologous temperature T/T_{solvus}.

Applicability of Classical Theory to Al-Sc Alloys Reacted at 573 K

In their treatment of nucleation in a nonuniform regular solution, Cahn and Hilliard [36] suggested

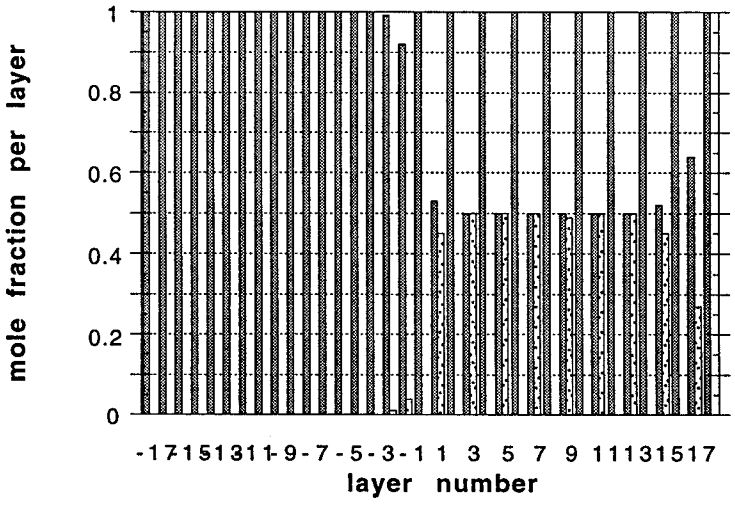
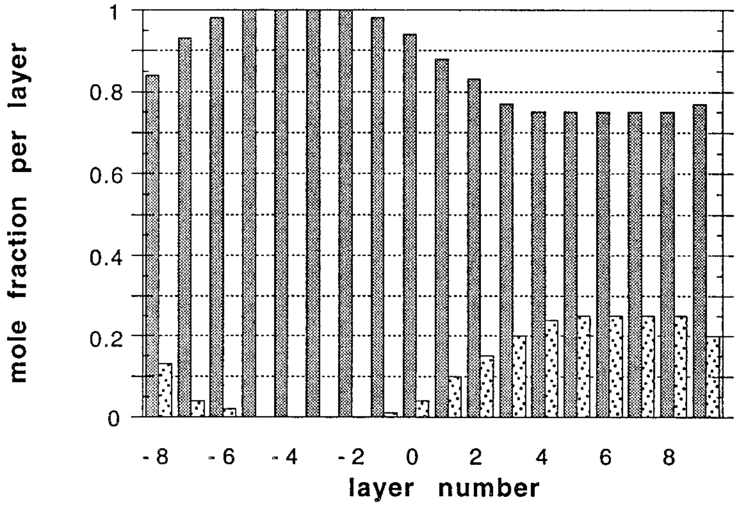
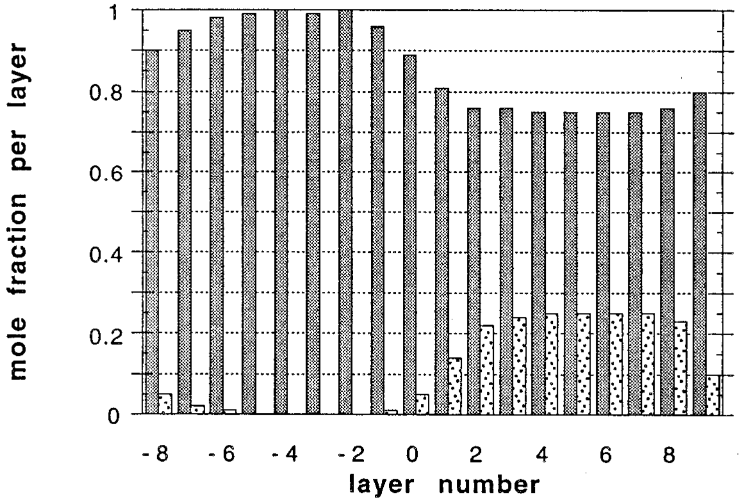


Figure 9. Larger system interphase (100) interface composition profile at 573 K.



a) Al(0.25%Sc):Al_3Sc



b) Al(1.0%Sc):Al_3Sc

Figure 10. Interphase (111) interface composition profiles for Al(0.25%Sc):Al_3Sc and Al(1.0%Sc):Al_3Sc at 573 K. Dark bars represent mole fraction of Al. Light bars represent mole fraction of Sc.

that a useful criterion for determining the applicability of classical theory can be expressed in terms of γ and the extent of the compositional diffuseness, l, as follows:

$$\frac{\pi\gamma l^2}{45\,k\,T} << 1 \qquad \text{(III.6)}$$

In this expression, k is Boltzmann's constant and T is absolute temperature. As long as the width of the diffuse region is small compared with the nucleus size, classical theory is expected to be a reasonable approximation. Taking the LTE-calculated value of γ at 573 K, ~32 mJ/m^2, and using a compositionally diffuse region of 4 • $d_{(100)}$ with a lattice parameter of 4.077 Å, the argument on the left side of equation (III.6) is approximately 0.75. This value is not small compared to unity. Alternatively, taking the average value of the interfacial energy extracted from experiment [3] and using the definition of the radius of a classical critical nucleus, r*, in terms of the interfacial energy and the strain energy corrected volume free energy change, the size of the nucleus can be calculated from equation (III. 7) and compared to the width of the diffuse region simulated in the present work:

$$r^* = \frac{-2\,\gamma}{\Delta F_V + \varphi} \qquad \text{(III.7)}$$

where ϕ is the elastic volume strain energy. Using values for the interfacial energy of 94 mJ/m^2 [3], for ΔF_v of -4.855 x 10^8 J/m^3, and for ϕ of 1.92 x 10^7 J/m^3, a value of r* = 4.07 Å is obtained. This value is roughly the unit cell dimension. (The larger of the two experimentally determined values for the interfacial energy was used because it should result in an upper bound for the nucleus radius.) This value of the nucleus radius is smaller than the calculated interfacial thickness, l. With respect to the Cahn-Hilliard criterion discussed above, it is unlikely that nucleation of Al_3Sc should be described classically in the temperature range examined in [3]. This may be anticipated in view of the large undercooling that was used in [3] to avoid the cellular reaction that occurs at temperatures above about 650 K. Using the phase disappearance method, a solvus temperature of about 840 K was secured for these alloys. This implies an undercooling of greater than 260 K for the experimental measurements of nucleation kinetics. Under such conditions, the classical definition given by equation (III.7) is not very physical. It is probably sufficient to interpret these findings as a strong indication that a wide range of embryo shapes may become stable nuclei and that analysis of the nucleation barrier heights is best undertaken in terms of nonclassical theory in this case.

Conclusions

The orientation dependence of the Al(fcc):$Al_3Sc(L1_2)$ interphase boundary energy (at 0 K) and the temperature dependence of γ were calculated using EAM based atomistic simulation and low temperature expansion methods, respectively. Atomistic 0 K results indicated that the (100) interface should be the most stable of the three orientations studied and that the (111) interface should be the least stable; thus, Al_3Sc precipitates at 0 K would be expected to be principally cuboidal. LTE calculations of the (100) interface showed that the interfacial energy was roughly temperature independent up to 400 K. Above 400 K, the interfacial energy was observed to decrease with increasing temperature as the entropy contribution became more significant with increasing interfacial compositional diffuseness. At 573 K, MC calculations showed a four atomic layer compositionally diffuse region was observed at the interphase interfaces. Independent estimates of the critical nucleus dimensions and the relative barrier heights to nucleation applicable at 573 K in Al-0.1 at%Sc indicate that critical nuclei of Al_3Sc probably are nonclassical and, therefore, are likely to be nonuniform during the early stages of precipitation. A nonclassical theory of nucleation of an ordered phase from a disordered solid solution would be required to make a more quantitative estimate of the properties of the nuclei at large undercoolings by means of continuum theoretical treatments.

IV. Thermodynamic Properties of Interphase Boundaries

In multiphase materials the interfacial layer between two coexisting phases is commonly referred to as an *interphase boundary* (IPB). The structural and thermodynamic properties of IPBs play an important role in controlling the morphology of precipitates as well as the kinetics of their formation. For example, the classical steady-state nucleation rate is related exponentially to the cube of the *interphase energy*, defined as the excess free energy per unit area associated with an IPB. In the absence of strain the equilibrium shape of a precipitate is determined by the dependence of the interphase energy upon crystallographic orientation. In alloys IPBs are generally compositionally diffuse [23], and it has been pointed out by Cahn and Hilliard [36] that in situations where the "width" of the IPB is on the order of the size of the critical radius, the thermodynamic formalism underlying classical nucleation theory is of questionable validity.

In this section results of recent calculations will be presented for the equilibrium structural and thermodynamic properties of coherent IPBs between phases with FCC-based crystal structures in the Ag-Al and Al-Li alloy systems. Specifically, in Ag-Al we consider IPBs between two disordered phases: the α-Al (solid solution) and ε-Guinier-Preston-zone (GP-zone) phases (see for example [46]). In Al-Li by contrast, calculations have been performed for IPBs between disordered α-Al and the *ordered* δ′ Al_3Li phase with an $L1_2$ crystal structure. The cluster variation method (CVM) [8, 47] and low-temperature-expansion (LTE) [45] statistical-mechanical techniques have been used to calculate interphase energies and equilibrium composition profiles as a function of temperature for IPBs with high-symmetry crystallographic orientations. The energetic parameters required as input for these calculations have been obtained in two ways: in the case of Al-Li they were taken from the phase-diagram-fitting work of Garland and Sanchez [48], whereas for Ag-Al they were derived from the results of first-principles total energy calculations. The results presented below for Ag-Al provide an example demonstrating how ab-initio techniques can be applied to the study of finite-temperature IPB properties in alloys.

Computational Approach

Let $\Omega(\Delta\mu, T)$ denote the value of the *grand potential* [44] in a binary alloy at temperature T and *chemical field* (defined here as half the difference between the chemical potentials for the two atomic species) $\Delta\mu$. Ω_0 and $\Delta\mu_0$ will denote the values of the grand potential and chemical field corresponding to bulk thermodynamic equilibrium between the two alloy phases of interest. Additionally, $\Omega_{hkl}(\Delta\mu_0, T)$ will denote the value of the grand potential for an inhomogeneous system containing separate spatial regions of these two phases together with coherent IPBs between them oriented along (hkl). If one can calculate $\Omega_{hkl}(\Delta\mu_0, T)$ and $\Omega_0(\Delta\mu_0, T)$ a value of the interphase energy γ_{hkl} can be derived from the following equation [44]:

$$\Omega_{hkl}\left(\Delta\mu_0, T\right) - \Omega_0\left(\Delta\mu_0, T\right) = \gamma_{hkl} A_{hkl} \qquad \text{(IV.1)}$$

where A_{hkl} is the total cross-sectional area associated with the IPBs.

For the purpose of calculating the values of the various thermodynamic potentials entering Eq. (IV.1), the CVM [8, 47] and LTE [45] techniques have been used in the present study. In the CVM variational estimates of grand potentials are determined from minimizations of a free energy functional which explicitly takes into account contributions to the enthalpy and entropy arising from correlation between atoms within some "maximal" cluster of lattice points. The CVM free energy functional can be written in terms of a linearly independent set of *correlation functions* [47] defined as generalized short-range order parameters for multisite clusters. At low temperatures minimization of the CVM free energy functional becomes numerically ill-conditioned due to the fact that the entropy is expressed in terms of logarithms of cluster probabilities, the values of which approach zero or one at low *T*. At these low temperatures values of the grand potential can be calculated using the LTE technique in which a Taylor series expansion of the logarithm of the partition function is performed about the zero-temperature ground-state energy [45].

The details of the CVM and LTE calculations performed in the current study were given previously [49] and will be only briefly described here. The tetrahedron-octahedron approximation [47] of the CVM was employed and single and double-atom substitutional excitation energies were taken into account in the LTE. Values of Ω_0 and $\Delta\mu_0$ at a given temperature were determined from calculations of the grand potential as a function of chemical field for each of the bulk phases of interest; Ω_0 and $\Delta\mu_0$ are defined by the point where the Ω vs. $\Delta\mu$ curves for each of the two phases intersect. In order to compute the values of $\Omega_{hkl}(\Delta\mu_0, T)$ required to calculate the interphase energy according to Eq. (IV.1), CVM and LTE calculations were performed for large supercells [49] constructed to contain spatial regions of each of the bulk phases of interest together with (hkl)-oriented IPBs between them. Interesting structural information about the IPBs is provided by the values of the correlation functions obtained from CVM minimizations of the grand potential for the supercell. In particular, from the point and pair correlation functions the average concentration, long-range and short-range order parameter profiles can be calculated as a function of distance across IPBs.

In the CVM the enthalpy is written in terms of products between cluster correlation functions and *effective cluster interaction* (ECI) parameters [47]. The ECI parameters are formally defined as configurationally-averaged exchange energies [47] and their values can be calculated in a variety of ways [6, 50]. Empirical values of the ECIs can be derived by fitting calculated CVM free energies to experimentally measured thermodynamic and/or phase diagram data [50]. Alternatively, ECIs can be obtained by fitting the zero-temperature CVM expression for the energy to the results of first-principles total energy calculations performed for ordered alloy superstructures using any of a variety of quantum-mechanical, electronic-structure techniques [6]. In the current study empirical values of the ECI parameters for fcc-based alloys in the Al-Li system were taken from the phase-diagram-fitting work of Garland and Sanchez [48]. In the work of Garland and Sanchez the tetrahedron-octahedron approximation [47] of the CVM was used and values of the ECI parameters for first and second neighbor pairs were adjusted in order to obtain good agreement with experimental measurements for the α-δ′ phase boundaries (see references listed in [48]). For the current study of IPBs in Ag-Al values of the ECI parameters were derived from first-principles [51]. Specifically, zero-temperature total energy calculations were performed for 22 hypothetical fcc-based Ag-Al compounds using the linear-muffin-tin-orbital method in the atomic sphere approximation [52]. An excellent fit (to within a maximum error of less than 1 mRy/atom ≈ 1.3 kJ/mole) to these energies was obtained using an expression for the energy which included ECI parameters for 11 clusters containing as many as six atoms and possessing a range as far as the second-neighbor shell (these 11 clusters are the "subclusters" of the fcc regular tetrahedron and octahedron). The first-principles-derived ECI parameters for Ag-Al were used in a CVM calculation of the metastable fcc miscibility gap and the predicted critical temperature was found to agree to within 45 K of the experimentally measured value (715 K [46]).

Results for IPBs between α and δ′ Phases in Al-Li

In Figure 11 CVM and LTE results are presented for the interphase energies associated with IPBs between the α and δ′ phases in Al-Li. Figure 11 (a) displays the interphase energy plotted as a function of temperature for IPBs with three crystallographic orientations: {111} (dashed line, highest values of the interphase energy), {100} (solid line, lowest values of the interphase energy) and {110} (dashed-dotted line, intermediate values of the interphase energy). At low temperatures the calculated results display a large degree of crystallographic anisotropy which decreases as the temperature is increased. The interphase energies are found to be significantly temperature

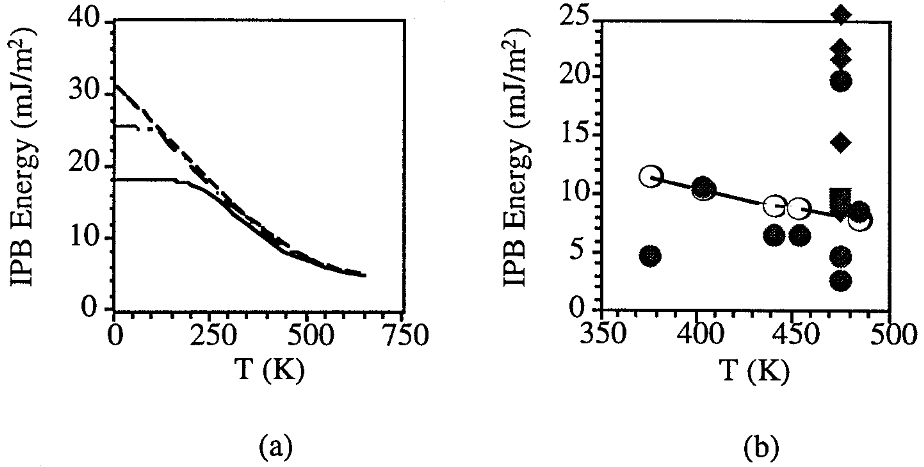


Figure 11. Calculated interphase energies for {100} (solid line, smallest values), {111} (dashed line, largest values), and {110} (dashed-dotted line, intermediate values) IPBs between α and δ' phases in Al-Li are plotted as a function of temperature in (a). In (b) the calculated results for {100} IPBs are plotted as open symbols. The filled circles, square and triangles in (b) correspond to values of the interphase energy based upon the analyses of experimental kinetic data performed by Hoyt and Spooner [53], Ardell [55] and Baumann and Williams [54], respectively.

dependent: the calculated values of γ decrease by as much as a factor of six when the temperature is raised from zero to 650 K. In the temperature range where δ' precipitates have been studied experimentally (between roughly 350 and 500 K) the interphase energies are found to be practically isotropic; this result is consistent with the fact that the shape of δ' precipitates has been observed to be roughly spherical. In Figure 11 (b) the results of the present calculations for the interphase energy of {100} IPBs (open circles) are compared with previously published [53-55] estimates (solid symbols) of the average interphase energy. The latter values, which are based upon analyses of experimentally measured nucleation and coarsening data, range from 2 to 25 mJ/m^2. In the present calculations the {100} interphase energy is predicted to vary from 7 to 11 mJ/m^2 over the temperature range displayed in Figure 11 (b). Generally speaking, it can be seen that the previously published estimates of the interphase energy bracket the present calculated values.

In Figure 12 the CVM-calculated composition profile is plotted across a {111} IPB at a temperature of 482 K. Each diamond symbol in Figure 12 corresponds to the average composition of a plane parallel to the IPB. The calculated compositions take on their values for the bulk α and δ' phases on the furthest left and right hand sides, respectively, of Figure 12. It can be seen that the distance over which the composition varies from its value in the α phase to that in the δ' phase across the IPB is sizable. From Figure 12 the width of the diffuse IPB is found to be roughly 6 lattice parameters or 24 Å. Calculations of the IPB width were performed at a number of temperatures between 373 K and 482 K. Over this temperature range the width was found to decrease monotonically with temperature to a value of 19 Å at 373 K.

Results for IPBs between α and ε-GP-Zone Phases in Ag-Al

In Figures 13 and 14 results of first-principles calculations are plotted for IPBs between the disordered α-Al and ε-GP-zone phases in the Ag-Al system. In Fig. 3 (a) calculated values of the

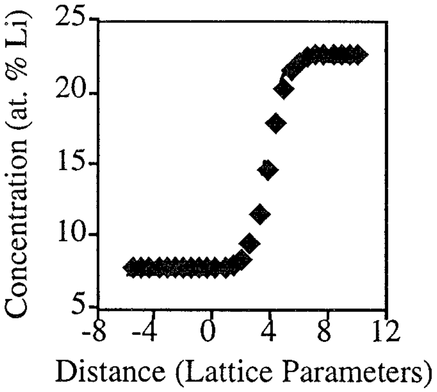


Figure 12: Calculated composition profile for a {111} IPB between α and δ′ phases in Al-Li at a temperature of 473 K. Each filled diamond symbol corresponds to the average composition (plotted in units of atomic % Li) in a plane parallel to the IPB. The horizontal axis corresponds to the distance perpendicular to the IPB.

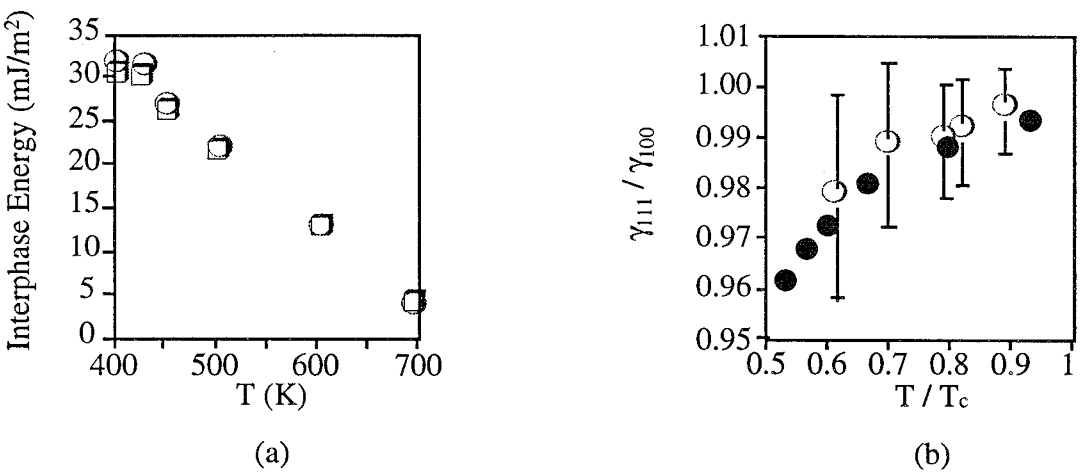
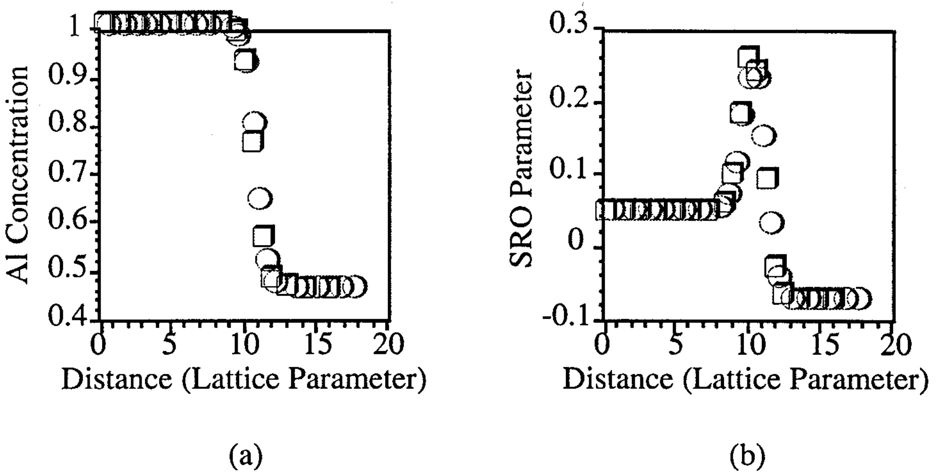


Figure 13: Calculated results for IPBs between α-Al and ε-GP-zone phases in Ag-Al. In (a) calculated interphase energies for IPBs with {111} and {100} orientations are plotted with open circles and squares, respectively. In (b) the filled circles denote calculated values of the ratio $\gamma_{111}/\gamma_{100}$ (taken from the results in (a)) plotted as a function of temperature divided by the critical temperature T_c. The open circles in (b) denote values of the ratio $\gamma_{111}/\gamma_{100}$ obtained from the Wulff construction using values of {111} and {100} facet lengths for ε-GP-zone precipitates measured by Alexander et al. [57] using transmission electron microscopy.



(a)

(b)

Figure 14: The calculated profiles for the composition and the nearest-neighbor, in-plane Warren-Cowley short-range-order parameter are shown in (a) and (b), respectively. The circle and square symbols correspond to results for {100} and {111} orientations, respectively. As in Fig. 2, the compositions are averaged over all sites in planes parallel to IPBs. Similarly, the short-range-order parameters are averaged over all nearest-neighbor pairs within planes parallel to interphase interfaces.

interphase energy are plotted against temperature for IPBs with {100} (open circles) and {111} (open squares) crystallographic orientations. It can be seen that the calculated values of γ monotonically decrease as a function of temperature; at the critical temperature corresponding to the top of the metastable fcc miscibility gap the interphase energies vanish for all crystallographic orientations [23]. The values of γ for {111} IPBs are lower than those for {100}-oriented interfaces at all temperatures. This finding is consistent with the results of regular-solution-model [56] calculations which predict that IPBs with {111} orientations have the lowest values of the interphase energy at all temperatures in model phase-separating fcc alloy systems where the energetics are parametrized by a single nearest-neighbor-pair interaction.

In Figure 13 (a) it can be seen that the ratio $\gamma_{111}/\gamma_{100}$ decreases with increasing temperature. In Figure 13 (b) the first-principles-calculated values of this ratio are plotted (with filled symbols) as functions of the temperature divided by the critical temperature (T_c). Also plotted in Figure 13 (b) are values of the ratio $\gamma_{111}/\gamma_{100}$ derived from the results of transmission-electron-microscopy (TEM) measurements performed by Alexander et al. [57]. These latter values, which are plotted with open symbols, were computed through the use of the Wulff construction using the TEM-measured {100} and {111} facet lengths for ε-GP-zone precipitates. The error bars in Figure 13 (b) arise from the statistical variation associated with the measured precipitate facet lengths. The level of agreement found between the first-principles and TEM-based results shown in Figure 13 (b) is remarkable considering that the former were calculated from first-principles without the use of any adjustable parameters.

In Figure 14 CVM-calculated composition and short-range-order (SRO) parameter profiles are plotted for Ag-Al IPBs at a temperature of 450 K. The open circle and square symbols in Figure 14 (a) correspond to the average compositions of planes parallel to {100} and {111} IPBs, respectively, plotted as a function of distance across the interface. A monotonic decrease in the composition is predicted in going from the α-Al (left hand side of figure) to the ε-GP-zone (right hand side) phase. The calculated width of the IPB is seen to be approximately 4 lattice parameters

(16 Å). As the temperature is raised from 450 K the width of the IPB increases and diverges as the calculated critical point (T=760 K) is approached.

Each symbol in Figure 14 (b) gives a value of the nearest-neighbor Warren-Cowley SRO parameter averaged over all pairs *within* a plane parallel to the IPB. Positive values of the SRO parameters indicate a preference for like nearest-neighbor bonds, i.e., clustering SRO. It can be seen in Figure 14 (b) that an enhancement of the SRO parameter in the IPB region is predicted. In other words, within the IPB layer the tendency for local phase-separation is enhanced. This result can be understood as being a consequence of the fact that planes within the IPB layer have compositions which lie inside the bulk miscibility gap.

Interpretations of small-angle X-Ray scattering (SAXS) results [58] have led to suggestions that Ag segregates to IPBs between the matrix and ε-GP-zone precipitates in Ag-Al. Since such suggestions have been based upon somewhat indirect experimental observations, it is interesting to note that the calculated results in Figure 14 (a) provide no evidence for Ag segregation. In the current calculations for Ag-Al we have neglected the effect of local atomic displacements upon the alloy free energy. Presently work is under way to investigate if such displacive effects might lead to the type of non-monotonic composition profiles expected from experimental SAXS results.

Summary

Results of CVM and LTE calculations have been presented for the thermodynamic and structural properties of coherent IPBs in fcc substitutional alloy systems. It is demonstrated that the temperature dependencies of interphase energies and the widths of compositionally diffuse IPBs can be significantly large. Results presented for Ag-Al demonstrate how, through the combined use of quantum-mechanical total energy methods and accurate statistical-mechanical techniques, first-principles studies of IPB properties at finite temperatures can be performed. In the future such *ab-initio* approaches should prove useful for calculating IPB properties in situations where limited experimental information is available, as is often the case when metastable phases are involved and when the number of alloy constituents is greater than two.

V. Vacancy Dislocation Loops in EAM Aluminum

Vacancies play an important role in determining the kinetics of precipitation of second phases in metals. They both enhance solute diffusion and serve as building blocks for more complex defects, such as dislocation loops, that act as heterogeneous nucleation sites. In this work, we seek to determine the stability of vacancy clusters that act as sinks for single vacancies in aluminum. We use an embedded atom method (EAM) potential to determine the energetics of these clusters using both conjugate gradient minimization and molecular dynamics integration techniques. The results agree surprisingly well with an earlier near-neighbor broken bond model; both the EAM and broken-bond-model give rise to lower energies than predicted by elasticity theory for very small clusters. The energetics provided by this study will be used in a reaction rate theory based computational method constructed to explore precipitation mechanisms in metal alloys.

Vacancies cluster or uncluster to minimize the free energy of a metallic system. At low temperatures, vacancies cluster to minimize the system internal energy by reducing the number of unsatisfied atomic bonds, and at high temperatures they uncluster to maximize the entropy of the system. In this regard, vacancies can be considered to be similar to any minor constituent in a material where random dispersement in the major phase is not energetically favored -- a process of nucleation and growth of vacancy clusters occurs, within kinetic limits, as the temperature of the system is decreased [59]. The lowest energy state is to nucleate and grow these clusters on existing, preferentially external (and therefore, convex) surfaces. However, kinetic limits may cause nucleation of vacancies throughout the material, which subsequently undergo coarsening, the growth of larger clusters at the expense of smaller ones, to minimize the total system energy.

Vacancies can be viewed in two ways. The first is that they represent open volumes within a lattice that must be bounded by surfaces with the rest of the lattice. The second is that they represent defects within the lattice to which the lattice may conform so as to minimize the total system energy. For isolated vacancies or for small vacancy clusters, the cost of unsatisfied bonds is small compared to the cost of the strain energy that would be required to satisfy missing bonds, so the vacancy energy is minimized by ordering vacancies in a roughly spherical volume, as is commensurate with the first representation. However, as the size of the vacancy cluster increases, the number of unsatisfied bonds that can be satisfied by conforming the lattice to the defect increases while the strain energy of the lattice to conform to the defect decreases because the resulting strain field becomes more diffuse. At some size, vacancy clusters can favorably collapse to form vacancy dislocation loops that are composed of an edge dislocation coupled with a stacking fault. If the stacking fault energy is high, as is the case in Al, the loop may undergo an additional reaction to create a perfect unfaulted loop [60]. Irrespective of the loop character, the effects of the open volume associated with the vacancies is secondary to the effects of the lattice defects caused by larger vacancy clusters, and the second representation becomes more appropriate.

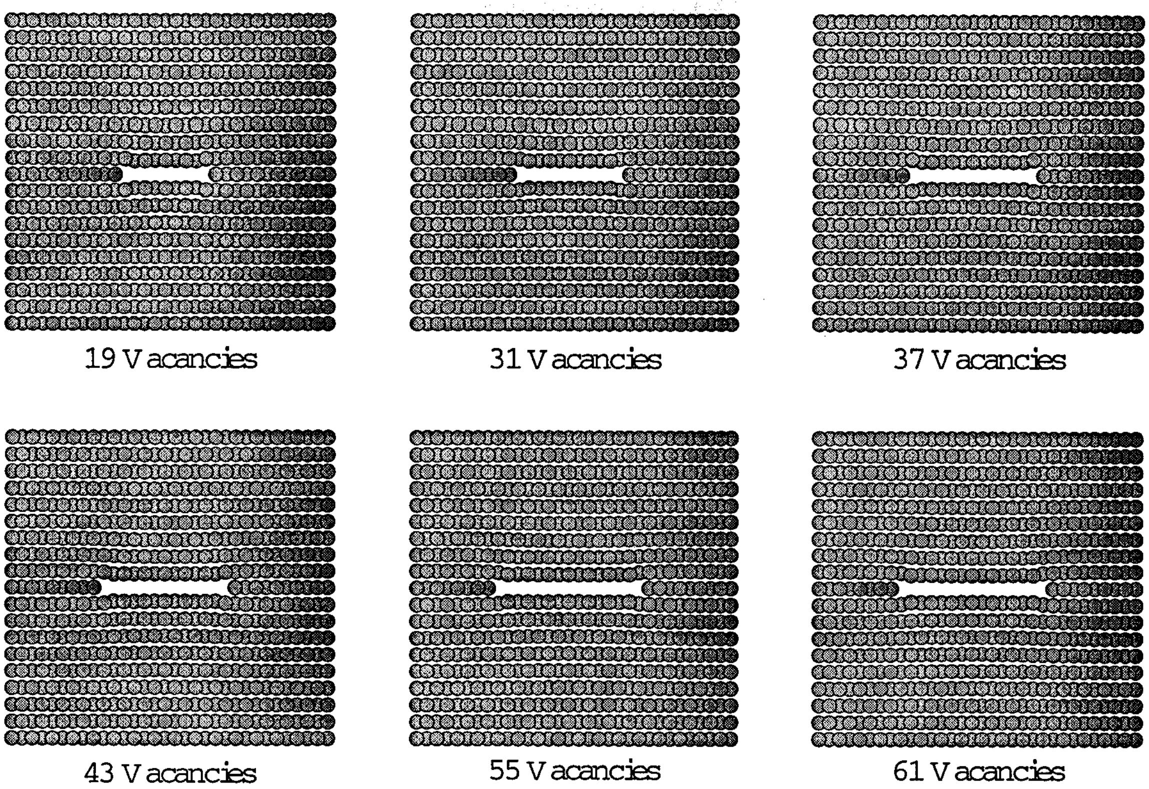
This work investigates the energetics of open versus collapsed vacancy clusters. We used the Dynamo program developed at Sandia National Laboratories to find the relaxed energies of lattices containing the various defects. The embedded atom method (EAM) [4] potential of Ercolessi [61] was used as this potential was specifically fit to the experimental stacking fault energy and surface energies of fcc Al. These energies are thought to play the most important role in establishing the stable defect structures. The intent of the study was to provide calculated formation energies of the vacancy clusters and a mechanistic understanding of their formation process for the reaction-rate-theory based code developed to model precipitation in Al alloys.

Procedure

We investigated the structures obtained by various routes of vacancy additions and minimization techniques. We first created vacancy clusters by sequentially removing the atoms having the highest individual potential energy within an existing lattice in a system which had been relaxed by a conjugate gradient method. We initially found significant atomic relaxations in the EAM aluminum when using this method of void formation for very limited numbers of vacancies, and we were interested in whether a vacancy dislocation loop would result when enough vacancies were added. However, we generally obtained roughly spherical voids exhibiting some surface relaxation.

We then created disk-shaped voids on a {111} plane to try to reproduce the experimentally verified vacancy dislocation loops found on these planes. The base system was a 5760-atom fcc aluminum structure oriented so as to make the {111} crystallographic plane parallel to the x-y simulation plane, resulting in a 49.4 x 45.6 x 41.9 Å periodic cell. Disk shaped voids were created by removing atoms to eliminate completely coordination shells about a central atom in a {111} plane -- 7, 13, 19, 31, 37, 43, 55, and 61 atoms. We then used two techniques to relax the resulting lattice -- the conjugate gradient minimization technique, and a molecular dynamics technique. The two methods gave significantly different relaxed structures.

It was found that the open voids were stable in the systems relaxed by the conjugate gradient method. To force the voids to collapse into vacancy dislocation loops, we applied an external stress perpendicular to the face of the voids. The stress required to collapse the voids was dependent on the size of the voids, and was the least for large voids and the greatest for small voids. We then removed the stress from the resulting structures to confirm that the collapsed structures were at least locally stable. Comparisons between the energy of the collapsed voids -- generally vacancy dislocation loops -- and the uncollapsed voids were made to show the more stable configuration. The voids readily collapsed in the structures relaxed using the molecular dynamics method, so no further application of pressure was necessary.



19 Vacancies

31 Vacancies

37 Vacancies

43 Vacancies

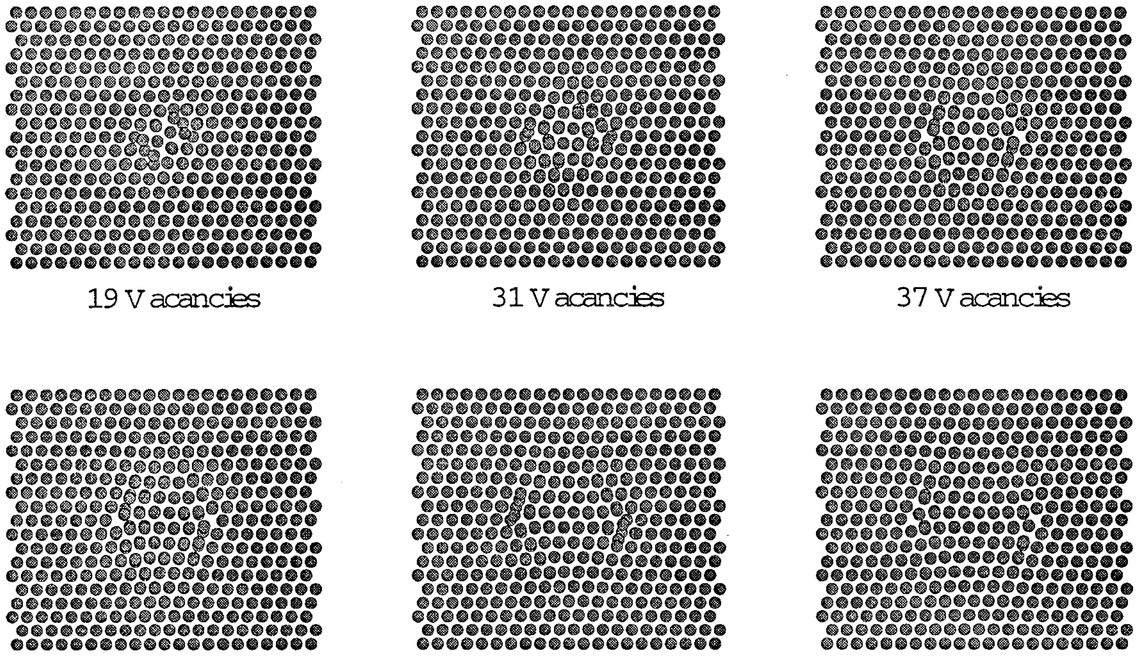
55 Vacancies

61 Vacancies

Figure 15. Structures of systems with uncollapsed disk-shaped voids on a (111) plane. View is ofa 6 angstrom-thick slice perpendicular to the plane of the voids.

TABLE IV: STRESS APPLIED AND SYSTEM RESPONSE DURING CONJUGATE GRADIENT MINIMIZATION

Number of Vacancies	Pressure, in order of application			
	1 kbar	10 kbar	100 kbar	20 kbar
7	None	None	-	-
13	None	None	Restructure	-
19	None	None	Collapse	-
31	None	None	Unstable	Collapse
37	None	None	Collapse	-
43	None	Collapse	-	-
55	None	Collapse	-	-
61	None	Collapse	-	-



19 Vacancies

31 Vacancies

37 Vacancies

43 Vacancies

55 Vacancies

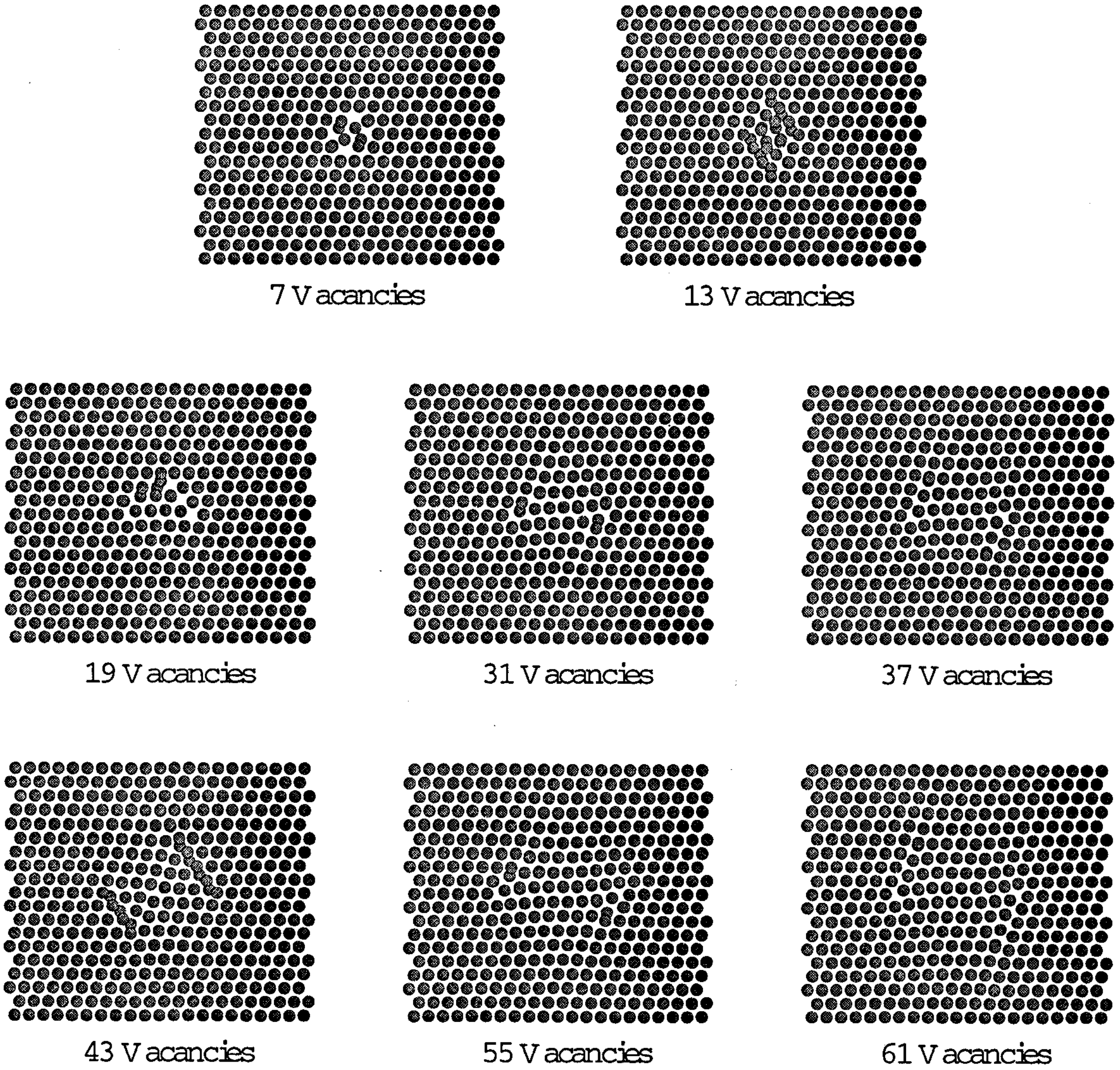
61 Vacancies

Figure 16. Structures of systems with collapsed and subsequently relaxed disk-shaped voids on a (111) plane. View is of a 6 angstrom-thick slice perpendicular to the plane of the voids.

We performed MD simulations at nearly 0K and at 300K. In order to compare the final potential energies, we made the assumption that in a thermally active harmonic system, the increase in potential energy resulting from thermal vibrations of the lattice is the same as the kinetic energy of the atoms in the lattice, in accordance with the equipartition theorem. Thus, we can obtain comparable potential energies by subtracting the kinetic energy of the systems from their potential energies. We performed a test of this method and found that this method yielded a correction within 3% of the actual correction obtained by comparing the potential energy of a system at 300K to that of a system at 0K. In our systems, this equates to a correction on the order of 20 eV and an error on the order of 0.5 eV.

We first used the final collapsed structures obtained from the energy relaxation simulations as starting structures for MD simulations at 300K. We will refer to this series of simulations as *series B*. At the conclusion of 2.0 ps simulations, we found that the voids did not reopen, but the strain field became more diffuse.

After finding that the collapsed voids were stable in the dynamic lattice, we then sought to determine if the open voids would spontaneously collapse in a dynamic lattice. We performed two series of simulations. First, we performed MD simulations at 300K using the relaxed un-collapsed vacancy clusters as the starting structures, which we will refer to as *series C*. Then we performed a series of constant energy MD simulations initially at 0K using the unrelaxed un-collapsed voids as the starting structures, which we will refer to as *series D*. In the latter case, an initial "kick" is given to the system as it relaxes to a local minimum energy state similar to that found by the conjugate gradient method, raising the temperature slightly and enabling further relaxations to occur. In almost all cases, the systems relaxed to collapsed voids, as shown in Figure 17 for *series D*.



7 Vacancies

13 Vacancies

19 Vacancies

31 Vacancies

37 Vacancies

43 Vacancies

55 Vacancies

61 Vacancies

Figure 17. Structures of systems with relaxed disk-shaped voids on a (111) plane using molecular dynamics. View is of a 6 angstrom-thick slice perpendicular to the plane of the voids.

The comparison of system energies for the collapsed and uncollapsed voids relative to that of a collection of individual relaxed vacancies (the cluster binding energy) is shown in Figure 18. Note that in all cases, the energies of the systems with collapsed voids are lower than corresponding systems with uncollapsed voids. We find little difference in the system energies as a result of different relaxation methods. There is some variation in the relative energies of the different methods for different sized vacancy clusters, mainly as a result of the scatter in the system energies of *series C* (MD system at 300K). The only series that is significantly different from the rest is *series B*, (MD relaxation of collapsed structure at 300K). We further investigated the cause of this difference.

It is known that dislocation loops can be either faulted or unfaulted -- that is, the collapsed walls may either meet at a stacking fault, or the dislocation loop may relax so as to eliminate this stacking fault via emission of a Shockley partial. Because the stacking fault energy in Al is quite high (150 mJ/m^2 [62]), we thought that the thermally active structures in *series B* could possibly have

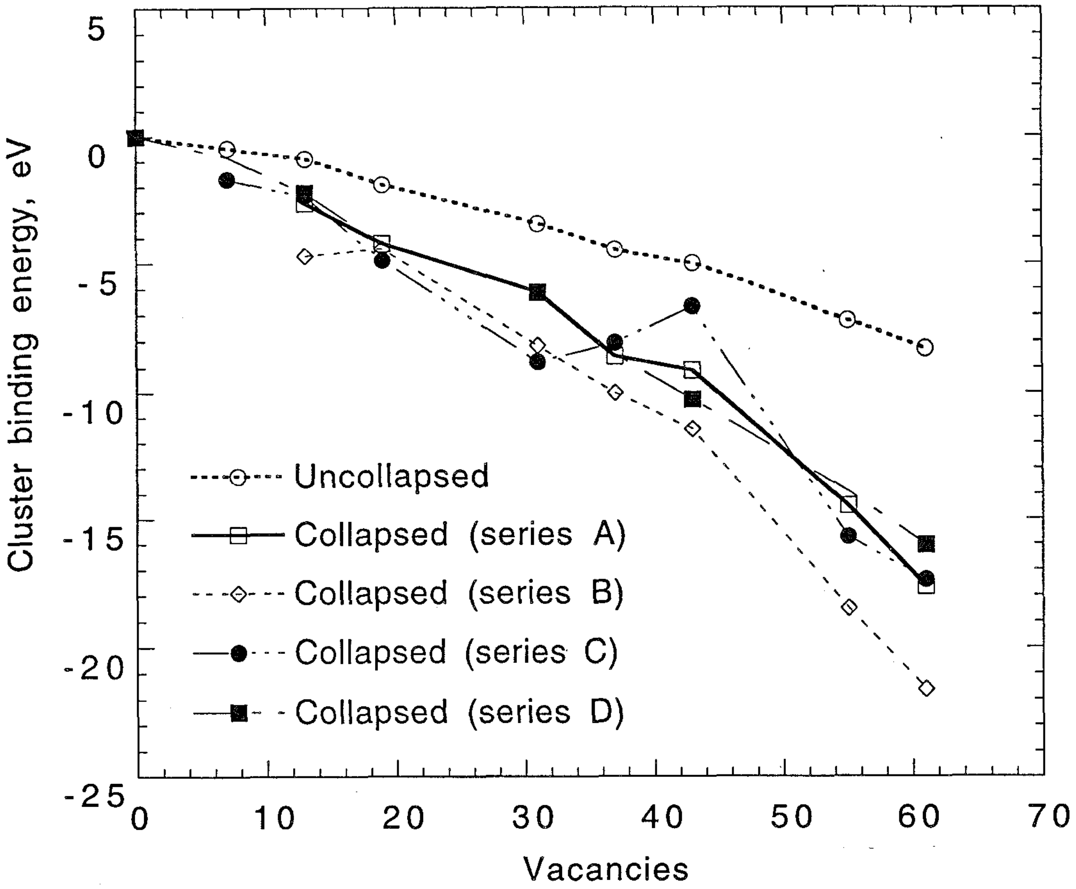


Figure 18: Comparison of vacancy cluster binding energies for uncollapsed disc-shaped voids and collapsed and relaxed voids.

undergone this reaction. Observation of the defect structures indicated that this was not the case and elasticity theory predictions support this result. The elastic criterion for the unfaulting reaction to occur is [60]

$$\gamma > \frac{Ga^2}{24\pi r}\left(\frac{2-\upsilon}{1-\upsilon}\right)\ln\left(\frac{2r}{r_0}\right) \tag{V.1}$$

where γ is the stacking fault energy (97 mJ/m^2 for the Al EAM potential used), G is the shear modulus (.332 x 10^{14} mJ/m^3 for the Al potential used), ν is Poisson's ratio (.319), *a* is the lattice parameter (4.05 x 10^{-10} m), r_0 is the dislocation core radius (taken as 5 x 10^{-10} m), and r is the loop radius. (Note that the elastic properties predicted by the Ercolessi potential were fitted to experimental values extrapolated to 0 K [60].) Using the Al properties listed in parentheses, the dislocation loop would have to have a radius of approximately 58 x 10^{-10} m before undergoing an unfaulting reaction.The largest loop radius we simulated was 12 x 10^{-10} m, smaller than this elastic criterion. It seems that the thermal activation simply allowed the defect structures to sample a broader configuration space than did the 0 K systems, thus allowing the series B structures to attain lower energies. It is also important to note that the atomistic calculations were carried out using periodic boundary conditions rather than boundaries fixed by the far-field elastic displacements resulting from the presence of the loop. The proximity of image dislocation loops may result in a change in the relative stability of the different dislocation loop structures.

Overall, the results clearly show that open disk-shaped vacancy clusters that are one atom-plane thick are not stable in a thermally active EAM aluminum lattice, and are only metastable in an energy-minimized static lattice. Differences in the relaxation methods result in only minor observable differences in the resulting structures.

Finally, the internal energies of formation (energy of the system containing the defect relative to the energy of a perfect crystal containing the same number of atoms) were calculated for both the open and collapsed structures. The formation energies were compared to near-neighbor broken bond formation energies of collapsed vacancy disks and elasticity theory calculations for dislocation loops possessing a Burgers vector normal to the loop plane.The former calculations were put forward by Davis [63] in a study of the nucleation of vacancy clusters in metals. He wrote the formation energy of the clusters as

$$\Delta F(i) = \left(3.0i + 3.53i^{1/2}\right)\frac{\Delta E_f}{6} - ikT\ln\left(\frac{n_v}{n_{v_s}}\right) \tag{V.2}$$

where i is the number of vacancies, ΔE_f is the formation energy for one vacancy, and n_v / n_{v_s} is the vacancy supersaturation ratio. Increasing the vacancy supersaturation counteracts the internal energy, ΔE, necessary to break atom bonds and form the vacancy defect. Thus at a given temperature, there exists a critical size vacancy disk which exists in metastable equilibrium with the vacancies in the metal matrix. The elasticity calculations were taken from Davis and Hirth [59] for a faulted loop as

$$\Delta E(i) = r\frac{Ga^3}{6(1-\upsilon)}\sqrt{\frac{3i}{4\sqrt{3}\pi}}\left[\ln\left(\frac{a}{r_0}\sqrt{\frac{3i}{4\sqrt{3}\pi}}\right) + 1.79\right] + \frac{3i}{4\sqrt{3}\pi}a^3\gamma \tag{V.3}$$

Figure 19 shows the internal energy of formation as a function of vacancies for the open vacancy disks, the collapsed dislocation loops, the near-neighbor broken bond model, and the elasticity calculations. It can be seen that the EAM results are quite similar to Davis' prediction for Al. The EAM results support Davis' conclusion that atomistic energies are necessary to accurately describe dislocation loops of small dimensions.

Conclusions

We were successful in forming vacancy dislocation loops in an EAM aluminum system. We have shown that these loops require some activation to form from a vacancy cluster, and that they remain stable in a thermally active system. The use of different methods for relaxing the lattices about the dislocation loops has little effect on both the final potential energy and structure of the resulting loops. We find that all loops are faulted despite variations in energies that would suggest that some loops are more relaxed than others, consistent with the elasticity theory based unfaulting criterion. Finally, we find that the nearest-neighbor bond model provides estimates in rough agreement with the potential energies of the vacancy clusters investigated using the many-body EAM potentials here.

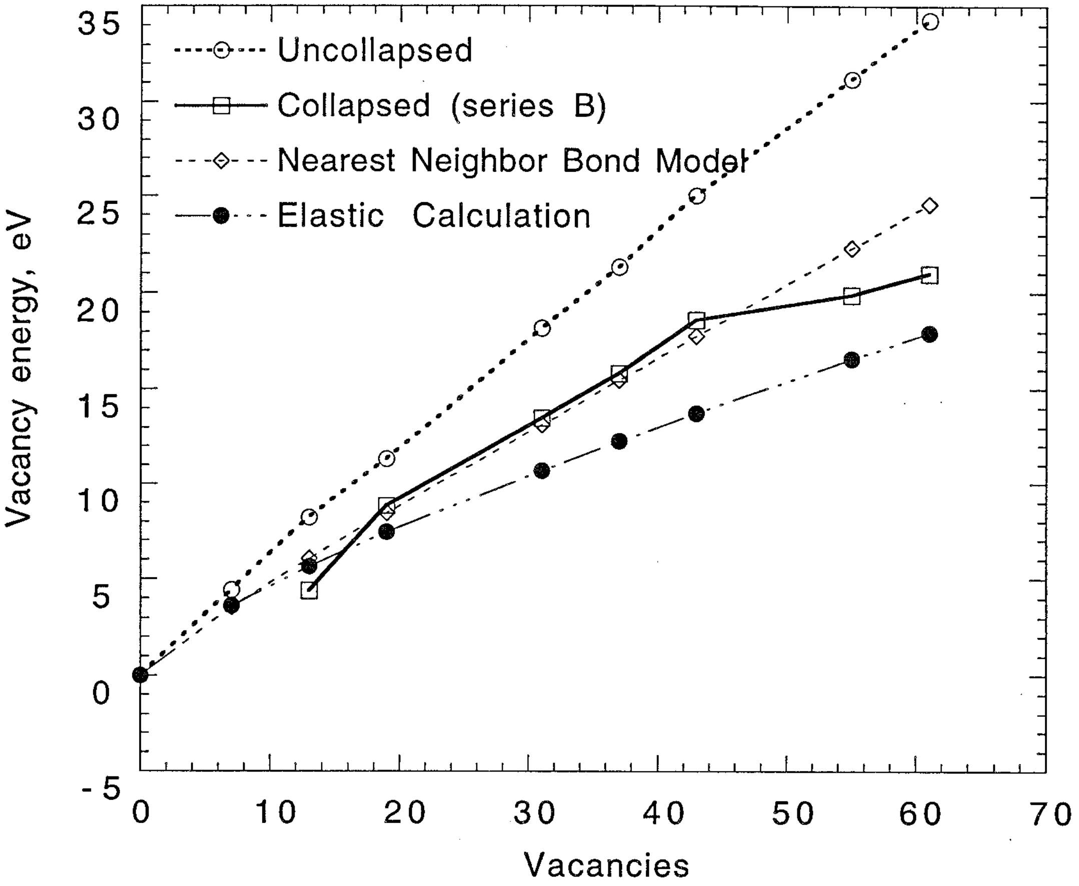


Figure 19. Comparison of vacancy cluster energies for uncollapsed disc-shaped voids and collapsed and relaxed voids.

VI. References

1. S. Glasstone, K. J. Laidler, H. Eyring, *Theory of Rate Processes* (McGraw-Hill, New York, 1941).
2. C. L. Rohrer, M. Asta, S. M. Foiles, R. W. Hyland, Jr., to appear in the proceedings of the *Thermodyanmics and Kinetics of Phase Transformations* symposium of the fall 1995 MRS meeting.
3. R. W. Hyland, Jr., C. L. Rohrer, S. M. Foiles and M. Asta, submitted to Acta Metallurgica.
4. M. S. Daw and M. I. Baskes, Phys. Rev. Lett. **50** (1983) 1285; M. S. Daw and M. I. Baskes, Phys. Rev. B **29** (1984) 6443.
5. R. W. Hyland, Jr., Metall. Trans. A **23** (1992) 1947.
6. D. de Fontaine, Solid State Physics **47**, 33 (1994); A. Zunger, in *Statics and Dynamics of Alloy Phase Transformations*, P. E. A. Turchi and A. Gonis, eds., NATO ASI Series, (Plenum, New York, 1994); F. Ducastelle, *Order and Phase Stability in Alloys,* North Holland, New York (1991).
7. M. Asta, to appear in the proceedings of the *Thermodyanmics and Kinetics of Phase Transformations* symposium of the fall 1995 MRS meeting.
8. R. Kikuchi, Phys. Rev. **81**, 998 (1951).
9. J. A. Venables, Phil. Mag. **27**, 697 (1973); J. B. Adams, W. N. G. Hitchon, L. M. Holzmann, J. Vac. Sci. Technol. A **6**, 2029 (1988); G. S. Bales and D. C. Chrzan, Phys. Rev. B **50**, 6057 (1994); J. H. Fikus and R. A. Johnson, Rad. Effects **40**, 63 (1979); M. F. Wehner and W. G. Wolfer, Phil. Mag. A **52**, 189 (1985); J. B. Adams and W. G. Wolfer, Acta metall. mater. **41**, 2625 (1993); K. Binder and D. Stauffer, Adv. in Phys. **25**, 343 (1976); K. F. Kelton and A. L. Greer, Phys. Rev. B **38**, 10089 (1988).
10. J. W. Gibbs, *Scientific Papers, Vol. I and II*, (Longmans Green, London, 1906); M. Volmer and A. Weber, Zeit. Phys. Chem. **119**, 227 (1926); L. Farkas, Zeit. Phys. Chem. **125**, 236 (1927); R. Becker and W. Döring, Ann. Phys. **24**, 719 (1935).
11. H. I. Aaronson and J. K. Lee, in *Lectures on the Theory of Phase Transformations,* edited by H. I. Aaronson (AIME, 1975), p. 87.
12. J. Feder, K. C. Russell, J. Lothe, G. M. Pound, Advances in Physics **15** (57), 111 (1966).
13. K. F. Kelton, A. L. Greer, C. V. Thompson, J. Chem. Phys. **79**, 6261 (1983).
14. K. C. Russell, in *Phase Transformations* (ASM Metals Park, OH, 1970), p. 2625.
15. C. P. Flynn, *Point Defects and Diffusion* (Clarendon Press, Oxford, 1972), p. 46.
16. C. Lane Rohrer, R. W. Hyland, Jr., S. M. Foiles, Report #93-56-EH12, Alcoa Technical Center, PA (1994).
17. R. Kampmann, Th. Ebel, M. Haese, R. Wagner, phys. stat. sol. (b) **172**, 295 (1992).
18. R. K. Trivedi, in *Lectures on the Theory of Phase Transformations,* edited by H. I. Aaronson (AIME, 1975), p. 79.
19. G. Sundar and J. J. Hoyt, J. Mater. Res. **10**, 1674 (1995).
20. R. W. Balluffi, J. Nucl. Mater. **69&70**, 240 (1978).
21. S. Fujikawa and S. Ushino, Kakuriken Kenkyu Hokoku (Tohoku Daigaku) **121**, 226 (1988).
22. J. L. Murray, Report #89-56-EH7, Alcoa Technical Center, PA (1989).
23. J. W. Cahn and J. E. Hilliard, J. Chem. Phys., **28** (2) (1958) 258.
24. F. K. LeGoues, H. I. Aaronson and Y. W. Lee, Acta Metall., **32** (10) (1984) 1845.
25. F. K. LeGoues, Y. W. Lee and H. I. Aaronson, Acta Metall., **32** (10) (1984) 1837.
26. H. I. Aaronson and F. K. LeGoues, Metall. Trans. A, **23** (1992) 1915.
27. L. A. Willey, United States Patent No. 3,619,181 (1971).
28. R. R. Sawtell and C. L. Jensen, Metall. Trans. A, **21** (1990) 421.
29. M. E. Drits, E. S. Kadaner, T. V. Dobatkina and N. I. Turkina, Russ. Metall., **4** (1973) 152.
30. M. E. Drits, L. B. Ber, Yu. G. Bykov, L. S. Toropova and G. K. Anast'eva, Fiz. Met. Metalloved, **57** (6) (1984) 118.

31. V. I. Yelagin, V. V. Zakharov, S. G. Pavlenko and T. D. Rostova, Fiz. Met. Metalloved, **60** (1) (1985) 88.
32. L. S. Toropova, Yu. G. Bykov, V. M. Lazorenko and Yu. M. Platov, Fiz. Met. Metalloved, **54** (1) (1982) 201.
33. R. Becker, Ann. Physik, **32** (1938) 128.
34. J. D. van der Waals, trans. by J. S. Rowlinson, J. Stat. Phys., **20** (2) (1979) 197.
35. C. J. Sparks, E. D. Specht, G. E. Ice, P. Zschak and J. Schneibel, Mat. Res. Soc. Symp., **213** (1991) 363.
36. J. W. Cahn and J. E. Hilliard, J. Chem. Phys., **31** (3) (1959) 688.
37. A. F. Voter and S.P. Chen, Mat. Res. Soc. Symp. Proc., **82** (1987) 175.
38. G. Simmons and H. Wang, *Single Crystal Elastic Constants and Calculated Aggregate Properties: A Handbook* (The M.I.T. Press, Massachusetts, 1971)
39. L. Kaufman and K. Bernstein, *Computer Calculation of Phase Diagrams* (Academic Press, NY, 1970); K. F. Michaels, W. Lange III, J. R. Bradley and HIA, Met. Trans., **6A** (1975) 1843.
40. R. W. Hyland, Jr. and R. C. Stiffler, Scripta Metall. et Mat., **25** (1991) 473.
41. C. L. Fu, J. Mater. Res., **5** (5) (1990) 971.
42. S. M. Foiles and M.S. Daw, J. Mater. Res., **2** (1987), 5.
43. S. M. Foiles and M. S. Daw, private communication.
44. J. W. Cahn, in *Interfacial Segregation*, edited by W. C. Johnson and J. M. Blakely (ASM, Metals Park, Ohio, 1977).
45. G. Parisi, *Statistical Field Theory* (Addison-Wesley, Menlo Park, CA, 1988).
46. A. J. McAlister, Bulletin of Alloy Phase Diagrams **8**, 526 (1987).
47. J. M. Sanchez and D. de Fontaine, Phys. Rev. B **17**, 2926 (1978); **21**, 216 (1980); **25**, 1759 (1982); J. M. Sanchez, F. Ducastelle and D. Gratias, Physica **128A**, 334 (1984).
48. J. S. Garland and J. M. Sanchez, in *Kinetics of Ordering Transformations in Metals*, ed. H. Chen and V. K. Vasudevan, (TMS, Warrendale PA, 1992), pg. 207.
49. M. Asta, to appear in the proceedings of the International Workshop on *the Theory and Application of the Cluster and Path Probability Methods* (Plenum, New York).
50. C. Sigli and J. M. Sanchez, Acta metall. **34**, 1021 (1986).
51. M. Asta and D. D. Johnson, unpublished.
52. O. K. Andersen, O. Jepsen and M. Sob, in *Electronic Band Structure and its Applications*, edited by M. Yussouff (Springer Lecture Notes, 1987).
53. J. J. Hoyt and S. Spooner, Acta. metall. **39**, 689 (1991).
54. S. F. Bauman and D. B. Williams, Scripta metall. **18**, 611 (1984).
55. A. J. Ardell, in *Phase Transformations '87*, edited by G. W. Lorimer (Institute of Metals, London, 1988).
56. Y. W. Lee and H. I. Aaronson, Acta Metall. **28**, 539 (1980).
57. K. B. Alexander, et al., Acta Metall. **32**, 2241 (1984).
58. Ph. A. Dubey, B. Schonfeld, and G. Kostorz, Acta Metall. **39**, 1161 (1991).
59. T. L. Davis and J. P. Hirth, J. Appl. Phys., 37 (1966) 2112.
60. D. Hull and D.J. Bacon, *Introduction to Dislocations* (Pergamon Press, New York, 1984).
61. F. Ercolessi and J. B. Adams, Europhysics Letters **26**, 583 (1994).
62. M. J. Mills and P. Stadelmann, Phil. Mag. A **60**, 355 (1989).
63. T.L. Davis, J. Appl. Phys. **38** [9], 3756-3760 (1967).

APPENDIX

In this appendix a listing is given of the rate-theory computer code. The code presented below is set up to model the precipitation of Al_3Sc in supersaturated Al-Sc alloys. However, the code is ready applicable to any binary alloy system for which the relevant thermochemical data are known. The primary function of the code below is to integrate the coupled set of differential equations given in Eq. (II.1). In the code there exists three different numerical techniques; any of the three can be used to integrate the differential equations, but we have found the variable-time-step, fourth-order Runga-Kutta to be the most robust. Many of the subroutines in the code below have been adapted from *Numerical Recipes* (W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in Fortran, The Art of Scientific Computing*, Cambridge University Press, Cambridge, MA, 1992) to which the reader is referred for detailed explanations. At the end of the section we also include a sample input file which is set up to perform a quench and age of an Al-0.11 at. % Sc alloy.

Listing of code

```
!!!!!!!!!!!!!!!!!!!!!!!     MAIN PROGRAM     !!!!!!!!!!!!!!!!!!


      program rate
c
c coauthored by M. D. Asta, S. Foiles, R. W. Hyland, Jr.,
c              K. B. Lippert, C. Lane Rohrer
c
c**********************************************
c     Solution of Master-Equations describing
c     time-evolution of distribution of precipitates
c     for a number of phases in a super-saturated
c     alloy
c**********************************************
c
       include 'common.h'
c
      call input
      call iniliz
      call cool
c
      end


c!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE INPUT     !!!!!!!!!!!!!!!!


      subroutine input
       include 'common.h'
c
      namelist /inpcard/ tinit,tfin,corat,tconc,tau,nit,
     $     maxtime,maxtau,tautest,imethod,
     $     nph,stoich,alat,volph,icoh,eccen,
     $     gamma,coh,gbe,gthick,gdiam,
     $     difc,em,e2m,ibin,nans,
     $     ev,difvac,evacm,difdivac,edivacm,
     $     sconf,evacb,
     $     dibind,
     $     rhod,resol
c
      open(unit=10,file='input_file',status='old')
      open(unit=20,file='output_file',status='unknown')
c
c     variables and units:
c
c
c     tinit,tfin=initial and final (quench) temperatures in K
c     corat=cooling rate (K/s)
c
c     tconc=total concentration of minority species (atomic conc)
c     conc=concentration of minority species monomers still
c         in solution
c
c     tau=time step (seconds)
c     nit=number of iterations
c     maxtime = simulation time limit to assure that changing timestep
c         value doesn't overrun desired length of thermal practice (seconds)
c     maxtau= maximum timestep code should ever take (seconds)
c     tautest = output information every tautest seconds
c
c     nph=number of phases to be considered (maximum value nphas)
c     stoich(i)=x where stoichiometry of phase i is a_xb (a,b=
c       majority and minority species, respectively).
c       NOTE:  stoich need not be an integer
c     alat=lattice constant of majority species (m)
c     volph(i)=volume per formula unit of phase i (m^-3)
c     eccen(i)= b/a ratio (note: eccentricity = sqrt(1 - (b/a)^2))
c               -> b/a = 1    -> sphere
c               -> b/a < 1    -> prolate
c               -> b/a > 1    -> oblate
c     icoh(i)=0 if phase i is coherent, 1 if phase i is
c           incoherent
c     gamma(i)=interphase interface energy (J/m^2) for phase i
c     coh(i)= coherency strain energy (J/m^3) for phase i
c
c     gbe(i)=grain boundary energy (per m^2) (zero for homogeneously
c          nucleating phase)
c     gthick = mean gb thickness (m)
c     gdiam=mean grain diameter (m)
c
c     rhod=dislocation density (1/m^2)
c     resol=resolution of experiment (in meters):
c          all clusters with diameter
c          greater than resol are termed a ppt. for the purposes of
c          computing moments of the distribution
c
c     difc=impurity diffusion constant prefactor (m^2/s)
c     em=activation energy for impurity migration by monovacancies (eV)
```

```
c     e2m = " by divacancies
c     difvac=monovacancy diffusion constant prefactor
c     evacm=activation energy for monovacancy migration
c     difdivac=divacancy diffusion constant prefactor
c     edivacm=activation energy for divacancy migration
c     ev=vacancy formation energy (eV)
c     sconf = vacancy cluster configurational prefactor
c     evacb = mono-,di-,trivacancy binding energy
c
c     ibin = identifies binary solute element for choosing thermo. data
c       ibin = 2 -> Sc
c     dibind = binding energy of solute to form formula unit dimer
c
c     nans = 0 for initial run, 1 for restart run
c
      read(10,inpcard)
      tfin=tfin-1.d-8
      tinit=tinit+1.d-8
c
c  convert eV -> J/mole
c
      em = em * 1.602d-19 * avogadro
      e2m = e2m * 1.602d-19 * avogadro
      evacm = evacm * 1.602d-19 * avogadro
      edivacm = edivacm * 1.602d-19 * avogadro
      ev = ev * 1.602d-19 * avogadro
      do 100 i=1,3
        evacb(i) = evacb(i) * 1.602d-19 * avogadro
 100  continue
      do 200 i=1,nph
        dibind(i) = dibind(i) * 1.602d-19 * avogadro
 200  continue
c
      if (nph.gt.nphas) then
        write(6,*) 'nph greater than parameter nphas'
        stop
      end if
      write(20,inpcard)
      radres=resol/2.d0
c
      return
      end


c!!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE INILIZ     !!!!!!!!!!!!!!!!


      subroutine iniliz
       include 'common.h'
c
      pi = 2.d0*asin(1.d0)
c
      do 5 i = 0,natom
        cbrts(i) = dfloat(i)**(1.d0/3.d0)
 5   continue
      do 10 i = 1,nvac
        delsqrt(i) = sqrt(dble(i)) - sqrt(dble(i-1))
 10   continue
      do 12 i = 1,nph
        epro1(i) = sqrt(1.d0 - eccen(i)*eccen(i))
        epro2(i) = epro1(i) / log( (1.d0-epro1(i))/(1.d0+epro1(i)))
        eobl1(i) = sqrt(eccen(i)*eccen(i) - 1.d0)
        eobl2(i) = eobl1(i) / asin( eobl1(i)/eccen(i) )
 12   continue
c
c     vol=volume per mole of solid-solution phase (m^3)
c     NOTE:  fcc structure assumed
c
c     initialize all values of c0 to zero (except for value
c     for monomer which is fixed by concentration)
c
c   c0 -> # solute atoms in s.s. / m^3
c   tconc -> total mole fraction solute
c   ssconc -> mole fraction solute in s.s.
c   cvac -> # vacancy clusters (1/m^3)
c
c     OR
c
c     read in values of c0 from ratres file
c
      if (nans.eq.1) then
         open (unit=24,file='ratres',form='unformatted')
        read (24) timeinit,tconc,ssconc,
     $       (critmin(j),driftcut(j),j=1,nph),
     $       (nat(j),j=1,nph),
     $       ((c0(i,j),i=1,nat(j)),j=1,nph),
     $       (cvac(i),i=1,nvac),(sumc0(i),i=1,nph)
        close(unit=24)
        a=alat+dadc(ibin)*ssconc
        vol=((a*a*a)/4.d0)*avogadro
      else
        timeinit = 0.0d0
        ssconc=tconc
        a=alat+dadc(ibin)*ssconc
        vol=((a*a*a)/4.d0)*avogadro
        do 30 i=1,nphas
          nat(i) = 10
          critmin(i) = 99999
          driftcut(i) = 99999
          sumc0(i)=0.d0
          do 40 j=2,natom
            c0(j,i)=0.d0
 40         continue
          c0(1,i)=tconc*avogadro/vol
 30       continue
        rt = 8.314d0*tinit
        cvac(1) =
     $       (3.d0 * exp(-ev/rt)) * avogadro/vol
```

```
      endif
c
      write(20,99)cvac(1)
 99   format('initial cvac = ',d17.10)
c
c initialize all growth and emission rates to zero
c
      do 80 i=1,nphas
        do 75 j=1,natom
          grow(j,i) = 0.d0
          emit(j,i) = 0.d0
 75     continue
 80   continue
      do 90 i=1,nvac
        growvac(i) = 0.d0
        emitvac(i) = 0.d0
 90   continue
c
c initialize average separation between dislocations
c
      rdist = sqrt(1.d0/pi/rhod)
c
      return
      end


c!!!!!!!!!!!!!!!!!!!!!!   SUBROUTINE COOL     !!!!!!!!!!!!!!!!!


      subroutine cool
      include 'common.h'
      real*4 tarraye(2),tarrayd(2)
      real*4 etime,dtime
c
c     iterate over time
c
c     every tautest seconds:
c       -determine the maximum number of atoms (nat)
c        for a given phase with appreciable weight
c       -compute the average size
c       -output the data
c
      itest = 1
      taut = tautest
      tempold = 0.0d0
      time = timeinit
      nreport = 10000
      deltime = time
      do 10 i=1,nit
        if (time .gt. timeinit+maxtime) goto 99
        call newnat
        call newc
        call tdcon(time,cvac)
        if (abs(tempold-temp).gt.1.d-6) then
```

```
          tempold = temp
        end if
        if (time.ge.timeinit+taut) then
          itest = itest + 1
          taut = dble(itest)*tautest
          call avgsz
          call out1(i)
        end if
        if (mod(i,nreport).eq.0) then
          deltime = time - deltime
          avedt = deltime/dble(nreport)
          deltime = time
           write(6,*)"i=",i," time=",time," ave dtime=",avedt
       endif
 10   continue
c
 99   call avgsz
      call out1(i)
      return
      end


c!!!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE NEWNAT     !!!!!!!!!!!!!!!!!


      subroutine newnat
       include 'common.h'
      data ifcall/1/
      save ifcall
      data nbuffer/10/,c0limit/1.d-1/
c.....
c     determine new value of nat(i) (maximum number
c     of atoms in phase (i) which have appreciable
c     weight
c.....
c.....nbuffer is the size of the buffer region past the last "significant"
c.....c0 value and c0limit is the value that determines what is significant
c.....
      do 10 j=1,nph
        if (ifcall.eq.1) then
          nstart = 1
        else
          nstart = nat(j) - nbuffer
        endif
        nattmp = nat(j) - nbuffer
        do 20 i=nstart,nat(j)
          if (c0(i,j).gt.c0limit/i) then
            nattmp=i
          else
            goto 15
          end if
 20     continue
 15     if (ifcall.eq.1) then
          nat(j) = nattmp + nbuffer
```

```
c.....
c.....clean up any "noise" that may be at the tail of the initial data
c.....
          do 17 i = nattmp+1,natom
            c0(i,j) = 0.0d0
 17        continue
        else
          nat(j) = min(max(nat(j),nattmp+nbuffer),natom)
          do 25 i = nattmp+1,nat(j)
            c0(i,j) = 0.0d0
 25        continue
        endif
c.....
        if (nat(j).ge.natom) then
          write(20,*) "warning!!! nat has reached natom for j=",j
          write(6,*) "warning!!! nat has reached natom for j=",j
          stop
        endif
 10   continue
      ifcall = 0
c     print *,'nat:',(nat(j),j=1,nphas)
      return
      end


c!!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE NEWC     !!!!!!!!!!!!!!!!


      subroutine newc
c
c**************************************************
c     integration of master equations for one time
c     step using adaptive fourth-order Runge-Kutta method
c
c     Found in Numerical Recipes, chapter 16.
c**************************************************
c for solute and vacancies:
c     c0(i,j)=concentration (per m^3) of particles of size
c            i for phase j
c
c
c     eps = error tolerance in rk routine
c     scalefactor(natom*nphas) = weights for error tolerance in rk routine
c       set so that error/new value .lt. eps
c     htry = step size in time to try and take
c     hdid = step size actually taken
c     hnext = step size suggested by rk routine for next time around
c
      include 'common.h'
      integer phaseindex
      dimension dc0dt(natph+nvac)
      dimension scalefactor(natph+nvac)
      dimension c0tmp(natph+nvac)
      data dc0dt/nvac*0.d0,natph*0.d0/
      data scalefactor/nvac*0.d0,natph*0.d0/
      data eps/1.d-1/
      save hnext
      external derivs
      parameter (tiny=0.1d0)
      parameter (tinynot=1.d7)
      do 10 i=1,nvac
       c0tmp(i)=cvac(i)
  10   continue
      do 15 j=1,nph
       phaseindex=natom*(j-1)+nvac
       do 17 i=1,nat(j)
        c0tmp(i+phaseindex)=c0(i,j)
  17    continue
  15   continue
      call derivs(time,c0tmp,dc0dt)
c
c     Set up initial h step
c
      if (time .eq. timeinit) then
        htry = tau
      else
        htry = hnext
      end if
c
c     Set up scaling factors and temp arrays for rkqs.
c
      do 20 i=1,nvac
        sctest = abs(htry*dc0dt(i))
        if (sctest .le. 1.d4) then
          scalefactor(i) = tinynot
        else
          scalefactor(i) = sctest
        endif
 20    continue
      ntotal = nvac
      do 22 j=1,nph
        phaseindex=natom*(j-1)+nvac
        ntotal = ntotal + nat(j)
        do 23 k=1,nat(j)
          sctest = abs(htry*dc0dt(k+phaseindex))
          if (sctest .le. 1.d4) then
            scalefactor(k+phaseindex) = tinynot
          else
            scalefactor(k+phaseindex) = sctest
          endif
 23      continue
 22    continue
c
      goto (1100,1200,1300) imethod
      stop 'imethod'
 1100 continue
c
c     Do one adaptive rk step
```

```
c
     call rkqs(c0tmp, dc0dt, natph+nvac, time,
    $     htry, eps, scalefactor,
    $     hdid, hnext, derivs)
     goto 1999
 1200 continue
c
c     take Bulirsch-Stoer step
c
     call bsstep(c0tmp, dc0dt, ntotal, time,
    $     htry, eps, scalefactor,
    $     hdid, hnext, derivs)
     goto 1999
 1300 continue
c.....
c.....take a simple finite difference step
c.....
     call findif(c0tmp,dc0dt,ntotal,time,htry,derivs)
     hdid = htry
     hnext = htry
     goto 1999
c.....
c.....end of "case" for different imethod
c.....
 1999 continue
     do 2220 i=1,nvac
       cvac(i) = c0tmp(i)
 2220 continue
     do 2000 j=1,nphas
       phaseindex=natom*(j-1)+nvac
       do 1800 i=1,nat(j)
         c0(i,j) = c0tmp(i+phaseindex)
 1800   continue
 2000 continue
c
     time = time + hdid
     tau = hdid
c
c    call to rcalc to determine new concentration,vol,
c    precipconc, and voltrans
c    voltrans=fraction of 1 m^3 occupied by precipitates of
c          all phases
c
     call rcalc(c0tmp)
c
c    chk:  checks how well the Runge-Kutta method conserves
c         the number of minority atoms
c
     chk = (tconc*avogadro/vol)
     iflag = 0
     do 70 i=1,nph
c don't double count monomer solute in ss calculation
        if (iflag .eq. 0) then
           chk = chk - c0(1,i)
```

```
          iflag = 1
        endif
        istop = min(driftcut(i),nat(i))
        do 68 j=2,istop
          chk = chk - c0(j,i)*dble(j)
 68     continue
        chk=chk-precipconc(i)
 70   continue
      chk=chk*vol/avogadro
c
      return
      end


c!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE TDCON     !!!!!!!!!!!!!!!!


      subroutine tdcon(tdtime,tdcvac)
       include 'common.h'
      dimension tdcvac(nvac)
c
c     calculate temperature given the time after quench assuming
c     exponential cooling with cooling rate given by _corat_
c
      temp=tfin+(tinit-tfin)*exp(-corat*(tdtime-timeinit)/(tinit-tfin))
      rt=8.314d0*temp
c
      if (nvac .ge. 2) then
        dcon = difc*(exp(-em/rt)*tdcvac(1)+
     $          exp(-e2m/rt)*tdcvac(2))*(vol/avogadro)/3.
      else
        dcon = difc*exp(-em/rt)*tdcvac(1)*(vol/avogadro)/3.
      endif
c
      cveq = 3.d0*exp(-ev/rt)*avogadro/vol
      dvac=difvac*dexp(-evacm/rt)
      ddivac=difdivac*dexp(-edivacm/rt)
c
      return
      end


c!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE AVGSZ     !!!!!!!!!!!!!!!!


      subroutine avgsz
       include 'common.h'
c
c      avsz(i)=average size of precipitates of phase i (m)
c      sumc0(i)=density of clusters greater than critical size
c
      dimension rdrs(3)
      rdrs(1) = radres
      rdrs(2) = 0.75d0*radres
     rdrs(3) = 0.50d0*radres
     do 10 i=1,nph
       tmp1=0.d0
       tmp2=0.d0
       tmp3=0.d0
       tot=0.d0
       avsztot(i)=0.d0
       do 20 j=2,nat(i)
         tot=tot+c0(j,i)
         avsztot(i)=avsztot(i)+rad(j,i)*c0(j,i)
         if (j.ge.critmin(i)) tmp3=tmp3+c0(j,i)
 20     continue
       sumc0(i)=tmp3
       if (tot.ne.0.d0) avsztot(i)=avsztot(i)/tot
       do 30 k=1,3
         do 25 j=2,nat(i)
           if (rad(j,i).ge.rdrs(k)) then
             tmp1=tmp1+c0(j,i)
             tmp2=tmp2+rad(j,i)*c0(j,i)
           endif
 25       continue
         totc0(i,k)=tmp1
         if (tmp1.ne.0.d0) then
           avsz(i,k)=tmp2/tmp1
         else
           avsz(i,k)=0.d0
         endif
 30     continue
 10   continue
c
     return
     end


c!!!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE OUT1      !!!!!!!!!!!!!!!!


     subroutine out1(itn)
      include 'common.h'
c
c     output formatted date to screen and file 5
c
     write(20,5) time,itn,temp,tau
      write(20,55) ssconc,dcon,(ceqbulk(i),i=1,nph)
     write(20,555) cvac(1),cvac(1)-cveq,cveq
     write(20,5555)chk
   5  format('time=',d12.6,'  # timesteps=',i10,'  T=',f10.3,
     ^'  tau=',d12.6)
 55   format('ssconc= ',d17.10,' dcon= ',d17.10,' ceqbulk= ',2d17.10)
 555  format('cvac= ',d17.10,' cvac-cveq= ',d17.10,' cveq= ',d17.10)
 5555 format('chk= ',d17.10)
     write(20,*)'solute in precipitates (past size) ',
    $     (driftcut(i),i=1,nph)
     write(20,6) (precipconc(i),i=1,nph)
```

```
  6  format(4(1x,d15.8),/,4(1x,d15.8))
      write(20,*)'average size of phases: (observable, total)'
       write(20,6) ((avsz(i,k),k=1,3),avsztot(i),i=1,nph)
      write(20,*)'# density of precipitates:  (observable, >crit)'
       write(20,6) ((totc0(i,k),k=1,3),sumc0(i),i=1,nph)
      write(20,7) (nat(i),i=1,nph)
  7  format(6(1x,i8))
 8    format(5(1x,d15.8))
      do 10 i=1,nph
         write(20,*)' phase # ',i
         write(20,8) (c0(j,i),j=2,50,2)
         write(20,8) (c0(j,i),j=60,400,10)
         write(20,8) (c0(j,i),j=500,5000,500)
         write(20,8) (c0(j,i),j=5000,natom,5000)
 10   continue
c
c  for graphing purposes:
      do 1100 i=1,nph
         write(65+i,66)i,time,ssconc*100.d0,(totc0(i,k),k=1,3),
     $       c0(critmin(i)+1,i),critsize(i),rstar(i),avsztot(i)
 1100 continue
 66    format(i5,6(1x,d12.6),i7,2(1x,d12.6))
c
       open(unit=8,file='ratres',form='unformatted')
      write(8) time,tconc,ssconc,
     $    (critmin(j),driftcut(j),j=1,nph),
     $    (nat(j),j=1,nph),
     $    ((c0(i,j),i=1,nat(j)),j=1,nph),
     $    (cvac(i),i=1,nvac),(sumc0(i),i=1,nph)
      close(8)
c
      return
      end


c!!!!!!!!!!!!!!!!!!!!!!    SUBROUTINE DERIVS    !!!!!!!!!!!!!!!!


      subroutine derivs(t,clong,dc0dt)
       include 'common.h'
      integer phaseindex
      dimension clong(natph+nvac),dc0dt(natph+nvac),cvactmp(nvac)
      save
      data icall/0/
      icall = icall + 1
      do 2 i=1,nvac
        cvactmp(i)=clong(i)
  2   continue
      call rcalc(clong)
      call tdcon(t,cvactmp)
      call getflux
      call eqconc
      call growc(clong)
c
```

```
c     from growth and emission rates, as well as
c     current values of c0, calculate
c     the time derivative of c0
c note that grow terms contain monovacancy concentration
c
      totc01=-2.d0*clong(nvac+1)*grow(1,1)
c
      do 10 i=1,nph
        phaseindex = natom*(i-1)+nvac
        totc01=totc01 +
     $      clong(phaseindex+2)*emit(2,i)
        do 20 j=2,nat(i)-1
          dc0dt(phaseindex+j) =
     $          -clong(phaseindex+j) * (grow(j,i) + emit(j,i)) +
     $          clong(phaseindex+j-1) * grow(j-1,i) +
     $          clong(phaseindex+j+1) * emit(j+1,i)
          totc01 = totc01 +
     $          clong(phaseindex+j) * (emit(j,i)-grow(j,i))
 20     continue
c.....treat j=nat(i) separately so we don't use undefined values
c.....nothing is allowed to happen with clusters of size nat+1
        dc0dt(phaseindex+nat(i)) =
     $      -clong(phaseindex+nat(i)) * emit(nat(i),i) +
     $      clong(phaseindex+nat(i)-1) * grow(nat(i)-1,i)
        totc01 = totc01 +
     $      clong(phaseindex+nat(i))*emit(nat(i),i)
 10   continue
c
c set monomer rates of each phase equal to total rate of
c  monomers gained/lost to all phases
c
      do 45 i=1,nph
        dc0dt(nvac+natom*(i-1)+1) = totc01
 45   continue
        dc0dt(1) = -2.d0*clong(1)*growvac(1) - emitvac(1)
      return
      end


c!!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE RKQS      !!!!!!!!!!!!!!!!


       SUBROUTINE rkqs(y,dydx,n,x,htry,eps,yscal,
     $    hdid,hnext,derivs)
c
      include 'common.h'
      integer phaseindex
      dimension dydx(n),y(n),yscal(n),yerr(nvac+natph),
     ^ytemp(nvac+natph),yjunk(nvac+natph)
      EXTERNAL derivs
      PARAMETER (NMAX=natph+nvac)
CU    USES derivs,rkck
      PARAMETER (SAFETY=0.9d0,PGROW=-.2d0,PSHRNK=-.25d0,ERRCON=1.89d-4)
c
```

```
      h=htry
      if (h .gt. maxtau) h = maxtau
c debugging
c      write(18,*) h,' (y, dy/dt, ynew, err, err/scale)'
 1    call rkck(y,dydx,n,x,h,ytemp,yerr,derivs)
      errmax=0.d0
      do 11 j=1,nph
        phaseindex=natom*(j-1)+nvac
        do 11 i=1,nat(j)
          yjunk(i+phaseindex) =
     $        yerr(i+phaseindex)/yscal(i+phaseindex)
          errmax=max(errmax,abs(yjunk(i+phaseindex)))
 11   continue
      do 12 i=1,nvac
        yjunk(i) = yerr(i)/yscal(i)
        errmax=max(errmax,abs(yjunk(i)))
 12   continue
      errmax=errmax/eps
      if(errmax.gt.1.d0)then
        sf = SAFETY*(errmax**PSHRNK)
        if ( sf .lt. 0.1d0) then
          h = h*0.1d0
        else
          h = h*sf
        end if
        xnew=x+h
        if(xnew.eq.x)pause 'stepsize underflow in rkqs'
        goto 1
      else
        if(errmax .gt. ERRCON)then
          hnext=SAFETY*h*(errmax**PGROW)
        else
          hnext=5.d0*h
        endif
        hdid=h
        do 121 i = 1,nvac
          y(i)=ytemp(i)
 121    continue
        do 122 j = 1,nph
          phaseindex=natom*(j-1)+nvac
          do 123 i = 1,nat(j)
            y(phaseindex+i) = ytemp(phaseindex+i)
 123      continue
 122    continue
        return
      endif
      END


c!!!!!!!!!!!!!!!!!!!!!!    SUBROUTINE BSSTEP    !!!!!!!!!!!!!!!!


      subroutine bsstep(y,dydx,nv,x,htry,eps,
     ^yscal,hdid,hnext,derivs)
```

```
       include 'common.h'
      integer nv,nmax,kmaxx,imax,phaseindex
       real*8 eps,hdid,hnext,htry,x,dydx(nv),y(nv),yscal(nv),
     ^safe1,safe2,redmax,redmin,tiny,scalmx
      parameter (nmax=natph+nvac,kmaxx=8,imax=kmaxx+1,safe1=.25d0,
      ^safe2=.7d0,redmax=1.d-5,redmin=.7d0,tiny=1.d-30,scalmx=.1d0)
       integer i,iq,k,kk,km,kmax,kopt,nseq(imax)
       real*8 eps1,epsold,errmax,fact,h,red,scale,work,wrkmin,
     ^xest,xnew,a(imax),alf(kmaxx,kmaxx),err(kmaxx),
     ^yerr(nmax),ysav(nmax),yseq(nmax)
      logical first,reduct
       save a,alf,epsold,first,kmax,kopt,nseq,xnew
      external derivs
      data first/.true./,epsold/-1.d0/
       data nseq/2,4,6,8,10,12,14,16,18/
      if(eps.ne.epsold)then
        hnext=-1.d29
        xnew=-1.d29
        eps1=safe1*eps
        a(1)=nseq(1)+1
        do 11 k=1,kmaxx
          a(k+1)=a(k)+nseq(k+1)
11      continue
        do 13 iq=2,kmaxx
          do 12 k=1,iq-1
            alf(k,iq)=eps1**((a(k+1)-a(iq+1))/
     ^        ((a(iq+1)-a(1)+1.d0)*(2*k+1)))
12        continue
13      continue
        epsold=eps
        do 14 kopt=2,kmaxx-1
          if(a(kopt+1).gt.a(kopt)*alf(kopt-1,kopt))goto 1
14      continue
1       kmax=kopt
      endif
      h=htry
      do 15 j=1,nph
      phaseindex=natom*(j-1)+nvac
      do 15 i=1,nat(j)
        ysav(i+phaseindex)=y(i+phaseindex)
15    continue
      do 150 i=1,nvac
        ysav(i)=y(i)
150   continue
       if(h.ne.hnext.or.x.ne.xnew)then
        first=.true.
        kopt=kmax
      endif
      reduct=.false.
2     do 17 k=1,kmax
        xnew=x+h
        if(xnew.eq.x)pause 'step size underflow in bsstep'
         call mmid(ysav,dydx,nv,x,h,nseq(k),yseq,derivs)
        xest=(h/nseq(k))**2
        call pzextr(k,xest,yseq,y,yerr,nv)
        if(k.ne.1)then
         errmax=tiny
         do 16 i=1,nvac
          errmax=max(errmax,abs(yerr(i)/yscal(i)))
 16       continue
         do 160 j=1,nph
          phaseindex=natom*(j-1)+nvac
          do 160 i=1,nat(j)
           if(dabs(yscal(i+phaseindex)).ge.1.d-12) then
           errmax=max(errmax,abs(yerr(i+phaseindex)/
  ^         yscal(i+phaseindex)))
           end if
160       continue
         errmax=errmax/eps
         km=k-1
         err(km)=(errmax/safe1)**(1.d0/(2.d0*km+1.d0))
        end if
        if(k.ne.1.and.(k.ge.kopt-1.or.first))then
         if(errmax.lt.1.d0)goto 4
          if(k.eq.kmax.or.k.eq.kopt+1)then
           red=safe2/err(km)
           goto 3
         else if(k.eq.kopt)then
           if(alf(kopt-1,kopt).lt.err(km))then
            red=1.d0/err(km)
            goto 3
           endif
         else if(kopt.eq.kmax)then
           if(alf(km,kmax-1).lt.err(km))then
            red=alf(km,kmax-1)*safe2/err(km)
            goto 3
           end if
         else if(alf(km,kopt).lt.err(km))then
           red=alf(km,kopt-1)/err(km)
           goto 3
         end if
        end if
 17     continue
  3    red=min(red,redmin)
       red=max(red,redmax)
       h=h*red
       reduct=.true.
       goto 2
  4    continue
       hdid=h
       first=.false.
       wrkmin=1.d35
       do 18 kk=1,km
        fact=max(err(kk),scalmx)
        work=fact*a(kk+1)
        if(work.lt.wrkmin)then
         scale=fact
         wrkmin=work
        kopt=kk+1
       endif
  18  continue
      hnext=h/scale
       if(kopt.ge.k.and.kopt.ne.kmax.and..not.reduct)then
        fact=max(scale/alf(kopt-1,kopt),scalmx)
        if(a(kopt+1)*fact.le.wrkmin)then
          hnext=h/fact
          kopt=kopt+1
        endif
      endif
      return
      end


c!!!!!!!!!!!!!!!!!!!!!!    SUBROUTINE FINDIF    !!!!!!!!!!!!!!!!


       subroutine findif(y,dydx,n,x,h,derivs)
       include 'common.h'
      integer phaseindex
      dimension y(n),dydx(n)
c.....assume that derivs was called before the invocation here
      do 10 i = 1,nvac
        y(i) = y(i) + h*dydx(i)
  10 continue
      do 20 j = 1,nph
        phaseindex = natom*(j-1)*nvac
        do 30 i = 1,nat(j)
          y(i+phaseindex) = y(i+phaseindex) + h*dydx(i+phaseindex)
  30    continue
  20 continue
      return
      end


c!!!!!!!!!!!!!!!!!!!!!!    SUBROUTINE RCALC     !!!!!!!!!!!!!!!!


      subroutine rcalc(rc0)
       include 'common.h'
      integer which
      parameter (ntrial=100,tolx=1.0d-12,tolf=1.0d-12)
      dimension x(2)
c
      integer phaseindex
      dimension rc0(natph+nvac)
c
c     precipconc(i)=total number of minority species in
c                  ppts. of phase i in a volume of
c                  1 m^3
c     tcohpconc=total concentration of *atoms* in coherent
c             precipitates
c     volincp=volume occupied by incoherent ppts.
```

```
c
      tcohpconc=0.d0
      volincp=0.d0
      ss = 0.d0
      iflag = 0
      do 50 i=1,nph
        phaseindex=natom*(i-1)+nvac
        precipconc(i)=0.d0
c don't double count monomer solute in ss calculation
        if (iflag .eq. 0) then
          ss = ss + rc0(1+phaseindex)
          iflag = 1
        endif
c particles of size < r* + kT considered part of the solid solution
        if (driftcut(i) .ge. nat(i)) then
          do 55 j=2,nat(i)
            ss = ss + rc0(j+phaseindex)*dble(j)
 55       continue
        else
          do 56 j=2,driftcut(i)
            ss = ss + rc0(j+phaseindex)*dble(j)
 56       continue
          do 60 j=driftcut(i)+1,nat(i)
            precipconc(i)=precipconc(i)+
     ^          rc0(j+phaseindex)*dble(j)
 60       continue
        endif
        if (icoh(i).eq.1) then
          volincp=volincp+precipconc(i)*volph(i)
        else
          tcohpconc=tcohpconc+precipconc(i)*(stoich(i)+1.d0)
        endif
 50   continue
      tvcohp = tcohpconc
      tvincp = volincp
      sdadc = ss*dadc(ibin)
c
c     solve numerically for ssconc, using constrained
c     Newton-Raphson
c
c     this part is necessitated because ssconc is
c     a function of alat which is in turn a function
c     of ssconc
c
c     vol=volume per mole of solid-solution phase (m^3)
c     NOTE:  fcc structure assumed
c
      x(1) = alat
      which = 4
      call mnewt(ntrial,x,n,tolx,tolf,nphas,natom,which)
      alatss = x(1)
      a4 = (alatss*alatss*alatss/4.d0)
      ssconc = ss * a4 /
     $    (1.d0 - tvincp - (a4 * tvcohp))
```

```
      vol= a4*avogadro
c
c      volph(i)=volume per formula unit of phase i (m^3) from input
c      omega(i)=volume per formula unit of coherently strained
c            phase i (m^3) (volume taken from matrix vol/atom)
c
c       if phase is coherent, surface area assumed to be that
c       of replaced solid solution, if incoherent surface area
c       is that of transformed second phasec
c
c       rad(j,i)= 'a' axis of ellipsoidal cluster of phase i containing
c             j minority atoms (and j*stoich majority atoms) (unit=m)
c        note that if eccen(i)=1, 'a' axis reduces to radius of a sphere
c
      tmp2=1.d0/3.d0
      do 10 i=1,nph
        if (icoh(i).eq.0) then
          omega(i)= a4 * (stoich(i)+1.d0)
          tmp1=3.d0*omega(i)/(4.d0*pi*eccen(i)*eccen(i))
          tmp1 = tmp1**tmp2
          jj = mod(nat(i)+1,4)
          do 21 j=1,jj
            rad(j,i)=tmp1*cbrts(j)
 21         continue
          do 211 j=jj+1,nat(i)+1,4
            rad(j,i)=tmp1*cbrts(j)
            rad(j+1,i)=tmp1*cbrts(j+1)
            rad(j+2,i)=tmp1*cbrts(j+2)
            rad(j+3,i)=tmp1*cbrts(j+3)
 211        continue
        elseif (icoh(i).eq.1) then
          tmp1=3.d0*volph(i)/(4.d0*pi*eccen(i)*eccen(i))
          tmp1 = tmp1**tmp2
          do 20 j=1,nat(i)+1
            rad(j,i)=tmp1*cbrts(j)
 20         continue
        else
          write(6,*)'coherency of ppt phase ',i,' is unspecified'
          stop
        endif
c calculate areas and volumes for specified shapes from rad
c  sphere
        if (dabs(eccen(i)-1.d0).lt.1.d-10) then
          do 35 j=1,nat(i)+1
            a = rad(j,i)
            area(j,i) = 4.d0 * pi * (a*a)
            interfarea(j,i) = pi * (a*a)
            volume(j,i) = 4.d0*pi*a*a*a*eccen(i)*eccen(i)/3.d0
 35         continue
c  prolate
        elseif (eccen(i) .lt. 1.d0) then
          do 36 j=1,nat(i)+1
            a = rad(j,i)
            b = a*eccen(i)
            area(j,i) = 2.d0*pi *
     $          (b*b + a*b*asin(epro1(i)) / epro1(i))
c  note that a gb ppt would not form with this shape
            interfarea(j,i) = 0.0d0
            volume(j,i) = 4.d0*pi*a*a*a*eccen(i)*eccen(i)/3.d0
 36        continue
c  oblate
        elseif (eccen(i) .gt. 1.d0) then
          do 37 j=1,nat(i)+1
            a = rad(j,i)
            b = a*eccen(i)
            area(j,i) = 2.d0*pi*(b*b) +
     $        (pi*a*b / eobl1(i)) *
     $          log((eccen(i)+eobl1(i))/(eccen(i)-eobl1(i)))
            interfarea(j,i) = pi * (b*b)
            volume(j,i) = 4.d0*pi*a*a*a*eccen(i)*eccen(i)/3.d0
 37        continue
        endif
c
        do 40 j=2,nat(i)+1
          area2(j,i) = area(j,i) - area(j-1,i)
          vol2(j,i) = volume(j,i) - volume(j-1,i)
          interfarea2(j,i) = interfarea(j,i) - interfarea(j-1,i)
 40     continue
c
 10   continue
c
c set area available for attachment (vacflarea) for vacancy clusters assuming:
c 1) vacancy radius = atomic radius = rat,
c 2) vacancy clusters are disks on (111) planes with volume of i vacancies
c
      vat = vol/avogadro
      rat = (3.d0*vat/4.d0/pi)**tmp2
      rtmp = vat / (2.d0*pi) / rat
      do 70 i=1,nvac
        rvac(i) = sqrt(dble(i)*rtmp)
        vacflarea(i) = 2.d0*pi*rvac(i)*(2.0d0*rat)
 70   continue
c
      return
      end


c!!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE GETFLUX   !!!!!!!!!!!!!!!!


      subroutine getflux
       include 'common.h'
c
c prolate and oblate geometric factors are calculated in iniliz
c
       fourpi = 4.d0*pi
       do 10 i=1,nph
c prolate ellipsoid
```

```
        if (eccen(i) .lt. 1.0d0) then
          do 20 j=1,nat(i)
            a = rad(j,i)
            flux(j,i) = fourpi * (-2.d0*a) * epro2(i)
 20       continue
c oblate ellipsoid
        elseif (eccen(i) .gt. 1.0d0) then
          do 30 j=1,nat(i)
            a = rad(j,i)
            b = a*eccen(i)
            flux(j,i) = fourpi * b * eobl2(i)
 30       continue
c sphere
        elseif (dabs(eccen(i)-1.d0).lt.1.d-10) then
          do 40 j=1,nat(i)
            flux(j,i) = fourpi * rad(j,i)
 40       continue
        endif
 10   continue
      return
      end


c!!!!!!!!!!!!!!!!!!!!!!!    SUBROUTINE EQCONC    !!!!!!!!!!!!!!!!!


      subroutine eqconc
       include 'common.h'
      integer which
      parameter (ntrial=100,tolx=1.0d-12,tolf=1.0d-12)
      dimension x(2)
      save
      data temporig/0.d0/,ssold/0.d0/,iext/0/
c
      rt = 8.314d0*temp
      voloverav = vol/avogadro
      avovervol = avogadro/vol
      avoverrt = avogadro/rt
c
c if temperature or ssconc has not changed significantly or
c   # bins of ppts has not increased past where ceq has been calculated,
c   no need to recalculate thermodynamic quantities
c
      maxnat = 0.d0
      do 50 i=1,nph
        maxnat = max(maxnat,nat(i))
 50   continue
      if (abs(temp-temporig) .gt. 1.d-5 .or.
     $     abs(ssconc-ssold) .gt. 1.d-6 .or.
     $    maxnat .gt. iext) then
c
c check temperature change direction for establishing r*
c
        if (temp-temporig .gt. 1.d-5) then
        inc = 1
      elseif (abs(temp-temporig) .le. 1.d-5) then
        inc = 0
      elseif (temporig-temp .gt. 1.d-5) then
        inc = -1
      endif
      temporig = temp
      ssold = ssconc
      iext = min(maxnat+100,natom)
c
      do 100 i=1,nph
c
c calculate global equilibrium matrix concentration
c note that the ppt free energy is modified by the strain energy
c
        x(1) = ssconc*1.d-5
        cph = 1.d0/(stoich(i)+1.d0)
        x(2) = cph
        which = 1
        call mnewt(ntrial,x,n,tolx,tolf,i,natom,which)
        ceqbulk(i) = x(1)
c
c calculate free energies at equilibrium and current concentrations,
c derivatives of free energies at same, free energy of ppt
c
        if (icoh(i) .eq. 0) then
          vm = omega(i)
        elseif (icoh(i) .eq. 1) then
          vm = volph(i)
        endif
        call energy(i,vm)
c
c calculate critical radius
c (assumes stoichiometric ppt phase, but this can be adjusted...)
c
        gdenom = ( ghost - gppt(i) + dghost*(cph-ssconc) )
        if (abs(gdenom).le.1.d-6) then
          rstar(i) = 99999.d0
        else
          rstar(i) = (2.d0*gamma(i)*vm*avogadro*cph) / gdenom
        endif
c
c accommodate single phase region
c
        if (rstar(i) .gt. 0.d0) then
          crit = 4.d0*pi*rstar(i)*rstar(i)*rstar(i)/3.d0/vm
          if (crit .ge. 99999.d0) then
            critsize(i) = 99999
            idrftcut = 99999
          else
            critsize(i) = int(crit)
c calculate delta (x = delta/r*)
            which = 3
            x(1) = 0.1d0
```

```
                  call mnewt(ntrial,x,n,tolx,tolf,i,natom,which)
                  delta = x(1)*rstar(i)
                  cut = rstar(i) + delta
                  if (cut .ge. 99999.d0) then
                    idrftcut = 99999
                  else
                    idrftcut = int(4.d0*pi*cut*cut*cut/3.d0/vm)
                  endif
               endif
            else
              rstar(i) = 99999.d0
              critsize(i) = 99999
              idrftcut = 99999
            endif
            if (inc .eq. 0) then
              critmin(i) = min(critmin(i),critsize(i))
              driftcut(i) = min(driftcut(i),idrftcut)
            else
              critmin(i) = critsize(i)
              driftcut(i) = idrftcut
            endif
c
c calculate Gibbs-Thompson capillarity corrected matrix composition for
c detailed balance calculation for emission rates
c
            do 200 j=2,iext
              x(1) = ssconc*1.d-5
              x(2) = cph
              which = 1
              call mnewt(ntrial,x,n,tolx,tolf,i,j,which)
c Trivedi interfacial composition calculation
              ceq(j,i) = x(1)*avovervol
c Gibbs-Thompson simplified version for detailed balance expression:
              ceqgt(j,i) = (ceqbulk(i)*avovervol) *
     $              exp((gamma(i)*area2(j,i))
     $              * avoverrt )
 200          continue
c
 100  continue
      endif
      return
      end


c!!!!!!!!!!!!!!!!!!!!!    SUBROUTINE GROWC    !!!!!!!!!!!!!!!!


      subroutine growc(gc0)
      include 'common.h'
      integer phaseindex
      dimension gc0(nvac+natph),deltae(nvac+natph)
c
c    grow(i,j)=capture rate of solute atom by particle of i atoms
c           in phase j  (s^-1)
```

```
c     emit(i,j)=emission rate of solute atom "
c     growvac(i)=capture rate of monovacancy by cluster vacancy of size i
c     emitvac(i)=emission "
c
      rt = 8.314d0*temp
      dist = alat/sqrt(2.d0)
      freq = dcon*6.d0/(dist*dist)
      voloverav = vol/avogadro
      avovervol = avogadro/vol
      avoverrt = avogadro/rt
c
c  solute:
      do 10 i=1,nph
        phaseindex=natom*(i-1)+nvac
c
c set grain boundary site density - all sites available if gbe = 0
c if gb sites become full (sdens<0), no more precipitation at gb's
c
        if (dabs(gbe(i)).lt.1.d-10) then
          sdens = 1.0d0
        else
          sdens = ((gthick/gdiam)*avovervol - sumc0(i)) /
     $         (avovervol)
          if (sdens .lt. 0.d0) sdens = 0.d0
        endif
c
        cph = 1.d0/(stoich(i)+1.d0)
c
c special case for growth rate of monomers, emission rate of dimers
c factor of 2 for relative monomer diffusivity
c note grow redefined as g(i) * #monomers for efficiency in derivs
c
        grow(1,i) = 2.d0*freq*(area(2,i)/area(1,i))*sdens
        emit(2,i) = grow(1,i)*exp(-dibind(i)/rt)
        grow(1,i) = grow(1,i) * gc0(phaseindex+1)*voloverav
c
        if (driftcut(i) .lt. nat(i)) then
c
c for pre-critical embryos through critical + kT nuclei:
c random walk attachment and detachment
c at r*, grow*gc0(1)-emit should = 0
c
          do 20 j=2,driftcut(i)
            grow(j,i) = freq * (area(j+1,i)/area(1,i)) *
     $           exp( -( dgv(i)*vol2(j+1,i) +
     $           gamma(i)*area2(j+1,i) - gbe(i)*interfarea2(j+1,i))*
     $           avoverrt )
c Kelton, Greer, Thompson version - constrained equilibrium
            emit(j+1,i) = freq * (area(j+1,i)/area(1,i)) *
     $           gc0(phaseindex+1)*voloverav
            grow(j,i) = grow(j,i) * gc0(phaseindex+1)*voloverav
 20       continue
c
c for post-critical spherical nuclei:
```

```
c deterministic flux based upon quasi-stationary diffusion field using
c Gibbs-Thompson capillarity corrected matrix concentration at the interface
c
        do 30 j=driftcut(i)+1,nat(i)-1
          grow(j,i) = flux(j+1,i)*dcon*
     $         (ssconc - ceq(j+1,i)) /
c    $           (ssconc - ceqgt(j+1,i))
     $         (gc0(phaseindex+1)*voloverav - ceqgt(j+1,i))
          emit(j+1,i) = grow(j,i)*ceqgt(j+1,i)
          grow(j,i) = grow(j,i) * ssconc * avovervol
 30     continue
c
      else
c
        do 35 j=2,nat(i)-1
          grow(j,i) = freq * (area(j+1,i)/area(1,i)) *
     $         exp( -( dgv(i)*vol2(j+1,i) +
     $         gamma(i)*area2(j+1,i) - gbe(i)*interfarea2(j,i) ) *
     $         avoverrt)
          emit(j+1,i) = freq * (area(j+1,i)/area(1,i)) *
     $         gc0(phaseindex+1)*voloverav
          grow(j,i) = grow(j,i) * gc0(phaseindex+1)*voloverav
 35     continue
c
      endif
c
 10   continue
      growvac(nvac) = 0.d0
c
c emission from monovacancies = loss of monovacancies to dislocations +
c  loss of monovacancies to solute/vacancy pairs
c
      disloss = (2.d0*pi*rhod/log(rdist/(2.d0*alatss))) *
     $     dvac * (gc0(1)-cveq)
      emitvac(1) = disloss
c
      return
      end


c!!!!!!!!!!!!!!!!!!!!!!    SUBROUTINE RKCK     !!!!!!!!!!!!!!!!


      SUBROUTINE rkck(y,dydx,n,x,h,yout,yerr,derivs)
      include 'common.h'
      integer phaseindex
      dimension dydx(n),y(n),yerr(n),yout(n)
      EXTERNAL derivs
      PARAMETER (NMAX=natph+nvac)
CU    USES derivs
      dimension ak2(NMAX),ak3(NMAX),ak4(NMAX),ak5(NMAX),ak6(NMAX),
     *ytemp(NMAX)
      PARAMETER (A2=.2d0,A3=.3d0,A4=.6d0,A5=1.d0,A6=.875d0,B21=.2d0
     $     ,B31=3.d0/40.d0,
     *B32=9.d0/40.d0,B41=.3d0,B42=-.9d0,B43=1.2d0,B51=-11.d0/54.d0,
    $     B52=2.5d0,
     *B53=-70.d0/27.d0,B54=35.d0/27.d0,B61=1631.d0/55296.d0,
    $     B62=175.d0/512.d0,
     *B63=575.d0/13824.d0,B64=44275.d0/110592.d0,B65=253.d0/4096.d0,
    $     C1=37.d0/378.d0,
     *C3=250.d0/621.d0,C4=125.d0/594.d0,C6=512.d0/1771.d0,
    $     DC1=C1-2825.d0/27648.d0,
     *DC3=C3-18575.d0/48384.d0,DC4=C4-13525.d0/55296.d0,
    $     DC5=-277.d0/14336.d0,
     *DC6=C6-.25d0)
c
      do 110 i=1,nvac
       ytemp(i)=y(i)+B21*h*dydx(i)
110   continue
      do 115 i=1,nph
       phaseindex=natom*(i-1)+nvac
       do 115 j=1,nat(i)
        ytemp(j+phaseindex)=y(j+phaseindex)+
    ^    B21*h*dydx(j+phaseindex)
115   continue
      call derivs(x+A2*h,ytemp,ak2)
      do 120 i=1,nvac
       ytemp(i)=y(i)+h*(B31*dydx(i)+B32*ak2(i))
120   continue
      do 125 i=1,nph
       phaseindex=natom*(i-1)+nvac
       do 125 j=1,nat(i)
        ytemp(j+phaseindex)=y(j+phaseindex)+
    ^    h*(B31*dydx(j+phaseindex)+B32*ak2(j+phaseindex))
125   continue
      call derivs(x+A3*h,ytemp,ak3)
      do 130 i=1,nvac
       ytemp(i)=y(i)+h*(B41*dydx(i)+B42*ak2(i)+B43*ak3(i))
130   continue
      do 135 i=1,nph
       phaseindex=natom*(i-1)+nvac
       do 135 j=1,nat(i)
        ytemp(j+phaseindex)=y(j+phaseindex)+
    ^    h*(B41*dydx(j+phaseindex)+B42*ak2(j+phaseindex)+
    ^    B43*ak3(j+phaseindex))
135   continue
      call derivs(x+A4*h,ytemp,ak4)
      do 140 i=1,nvac
       ytemp(i)=y(i)+h*(B51*dydx(i)+B52*ak2(i)+B53*ak3(i)+B54*ak4(i))
140   continue
      do 145 i=1,nph
       phaseindex=natom*(i-1)+nvac
       do 145 j=1,nat(i)
        ytemp(j+phaseindex)=y(j+phaseindex)+
    ^    h*(B51*dydx(j+phaseindex)+B52*ak2(j+phaseindex)+
    ^    B53*ak3(j+phaseindex)+
    ^    B54*ak4(j+phaseindex))
145   continue
```

```
      call derivs(x+A5*h,ytemp,ak5)
      do 150 i=1,nvac
        ytemp(i)=y(i)+h*(B61*dydx(i)+B62*ak2(i)+B63*ak3(i)+B64*ak4(i)+
     *B65*ak5(i))
150   continue
      do 155 i=1,nph
        phaseindex=natom*(i-1)+nvac
        do 155 j=1,nat(i)
          ytemp(j+phaseindex)=y(j+phaseindex)+
     ^    h*(B61*dydx(j+phaseindex)+B62*ak2(j+phaseindex)+
     ^    B63*ak3(j+phaseindex)+
     ^    B64*ak4(j+phaseindex)+
     ^    B65*ak5(j+phaseindex))
155   continue
      call derivs(x+A6*h,ytemp,ak6)
      do 160 i=1,nvac
        yout(i)=y(i)+h*(C1*dydx(i)+C3*ak3(i)+C4*ak4(i)+C6*ak6(i))
160   continue
      do 165 i=1,nph
        phaseindex=natom*(i-1)+nvac
        do 165 j=1,nat(i)
          yout(j+phaseindex)=y(j+phaseindex)+
     ^    h*(C1*dydx(j+phaseindex)+
     ^    C3*ak3(j+phaseindex)+
     ^    C4*ak4(j+phaseindex)+
     ^    C6*ak6(j+phaseindex))
165   continue
      do 170 i=1,nvac
        yerr(i)=h*(DC1*dydx(i)+DC3*ak3(i)+DC4*ak4(i)+DC5*ak5(i)+DC6*
     *ak6(i))
170   continue
      do 175 i=1,nph
        phaseindex=natom*(i-1)+nvac
        do 175 j=1,nat(i)
          yerr(j+phaseindex)=
     ^    h*(DC1*dydx(j+phaseindex)+
     ^    DC3*ak3(j+phaseindex)+
     ^    DC4*ak4(j+phaseindex)+
     ^    DC5*ak5(j+phaseindex)+
     ^    DC6*ak6(j+phaseindex))
175   continue
      return
      END


c!!!!!!!!!!!!!!!!!!!!!!    SUBROUTINE MMID     !!!!!!!!!!!!!!!!


       subroutine mmid(y,dydx,nvar,xs,htot,nstep,
     ^yout,derivs)
       include 'common.h'
      integer nstep,nvar,nmax,phaseindex
      real*8 htot,xs,dydx(nvar),y(nvar),yout(nvar)
      external derivs
      parameter(nmax=natph+nvac)
      integer i,n
       real*8 h,h2,swap,x,ym(nmax),yn(nmax)
      h=htot/nstep
      do 11 i=1,nvac
        ym(i)=y(i)
        yn(i)=y(i)+h*dydx(i)
 11     continue
      do 110 j=1,nph
        phaseindex=natom*(j-1)+nvac
        do 110 i=1,nat(j)
          ym(i+phaseindex)=y(i+phaseindex)
          yn(i+phaseindex)=y(i+phaseindex)+
     ^     h*dydx(i+phaseindex)
110     continue
      x=xs+h
      call derivs(x,yn,yout)
      h2=2.*h
      do 13 n=2,nstep
        do 12 i=1,nvac
          swap=ym(i)+h2*yout(i)
          ym(i)=yn(i)
          yn(i)=swap
 12       continue
        do 120 j=1,nph
          phaseindex=natom*(j-1)+nvac
          do 120 i=1,nat(j)
          swap=ym(i+phaseindex)+
     ^     h2*yout(i+phaseindex)
          ym(i+phaseindex)=yn(i+phaseindex)
          yn(i+phaseindex)=swap
120       continue
        x=x+h
        call derivs(x,yn,yout)
 13     continue
      do 14 i=1,nvac
        yout(i)=0.5*(ym(i)+yn(i)+h*yout(i))
 14     continue
      do 140 j=1,nph
        phaseindex=natom*(j-1)+nvac
        do 140 i=1,nat(j)
        yout(i+phaseindex)=0.5*(ym(i+phaseindex)+
     ^  yn(i+phaseindex)+h*yout(i+phaseindex))
140     continue
      return
      end


c!!!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE PZEXTR    !!!!!!!!!!!!!!!!!


      subroutine pzextr(iest,xest,yest,yz,dy,nv)
       include 'common.h'
      integer iest,nv,imax,nmax,phaseindex
```

```
    real*8 xest,dy(nv),yest(nv),yz(nv)
    parameter(imax=13,nmax=natph+nvac)
    integer j,k1
    real*8 delta,f1,f2,q,d(nmax),qcol(nmax,imax),
   ^x(imax)
    save qcol,x
    x(iest)=xest
    do 11 j=1,nvac
      dy(j)=yest(j)
      yz(j)=yest(j)
 11   continue
    do 110 j=1,nph
      phaseindex=natom*(j-1)+nvac
      do 110 i=1,nat(j)
        dy(i+phaseindex)=yest(i+phaseindex)
        yz(i+phaseindex)=yest(i+phaseindex)
110   continue
    if(iest.eq.1)then
      do 12 j=1,nvac
        qcol(j,1)=yest(j)
 12     continue
      do 120 j=1,nph
        phaseindex=natom*(j-1)+nvac
        do 120 i=1,nat(j)
          qcol(i+phaseindex,1)=
   ^           yest(i+phaseindex)
120     continue
    else
      do 13 j=1,nvac
        d(j)=yest(j)
 13     continue
      do 130 j=1,nph
        phaseindex=natom*(j-1)+nvac
        do 130 i=1,nat(j)
          d(i+phaseindex)=
   ^          yest(i+phaseindex)
130     continue
      do 15 k1=1,iest-1
        delta=1./(x(iest-k1)-xest)
        f1=xest*delta
        f2=x(iest-k1)*delta
        do 14 j=1,nvac
          q=qcol(j,k1)
          qcol(j,k1)=dy(j)
          delta=d(j)-q
          dy(j)=f1*delta
          d(j)=f2*delta
          yz(j)=yz(j)+dy(j)
 14       continue
        do 140 j=1,nph
          phaseindex=natom*(j-1)+nvac
          do 140 i=1,nat(j)
          q=qcol(i+phaseindex,k1)
          qcol(i+phaseindex,k1)=
    ^       dy(i+phaseindex)
          delta=d(i+phaseindex)-q
          dy(i+phaseindex)=f1*delta
          d(i+phaseindex)=f2*delta
          yz(i+phaseindex)=
    ^      yz(i+phaseindex)+dy(i+phaseindex)
  140      continue
   15    continue
        do 16 j=1,nvac
         qcol(j,iest)=dy(j)
   16    continue
        do 160 j=1,nph
         phaseindex=natom*(j-1)+nvac
         do 160 i=1,nat(j)
         qcol(i+phaseindex,iest)=
    ^               dy(i+phaseindex)
  160    continue
       end if
       return
       end


c!!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE MNEWT     !!!!!!!!!!!!!!!!


       SUBROUTINE mnewt(ntrial,x,n,tolx,tolf,phn,atn,which)
       include 'common.h'
       integer atn,phn,which
       PARAMETER (NP=2)
CU     USES lubksb,ludcmp,usrfun="comtan"
       dimension fjac(NP,NP),fvec(NP),p(NP),x(NP)
c
       isig = 0
       cph = 1.d0/(1.d0+stoich(phn))
c
       do 14  k=1,ntrial
c
         if (which .eq. 1) then
           call comtan(x,n,NP,fvec,fjac,phn,atn,isig)
           if (isig .eq. 1) return
         elseif (which .eq. 3) then
           call getdelt(x,n,NP,fvec,fjac,phn)
         elseif (which .eq. 4) then
           call getalat(x,n,NP,fvec,fjac)
         endif
c
        errf=0.d0
        do 11 i=1,n
         errf=errf+dabs(fvec(i))
11       continue
        if(errf.le.tolf)return
        if (n .eq. 1) then
          p(1) = -fvec(1) / fjac(1,1)
        else if (n .eq. 2) then
```

```
          det = fjac(1,1) * fjac(2,2) - fjac(1,2) * fjac(2,1)
          p(1) = (-fvec(1)*fjac(2,2) + fvec(2)*fjac(1,2))/det
          p(2) = (-fvec(2)*fjac(1,1) + fvec(1)*fjac(2,1))/det
        else
          do 12 i=1,n
            p(i)=-fvec(i)
 12        continue
          call ludcmp(fjac,n,NP,indx,d)
          call lubksb(fjac,n,NP,indx,p)
        endif
        errx=0.d0
        do 13 i=1,n
         errx=errx+dabs(p(i))
cr make generic that x(i) cannot ever be negative
cr  (true for all solutions)
         fract1=1.d0
         fract2=1.d0
         if (p(i).lt.0.d0) fract2 = -x(i)/p(i)
cr assure that x(i) is never greater than concentration in second phase for
cr  capillarity corrected solution concentration
          if (which .eq. 1 .and. p(i).gt.0.d0) then
            fract2=(cph-x(i))/p(i)
         endif
cr assure that x(i) is never greater than alat +or- da/dc * tconc for
cr  composition dependent lattice parameter
         if (which .eq. 4) then
           dadct = dadc(ibin) * tconc
           if (p(i).gt.0.d0) then
             if (dadct .ge. 0.d0) then
               fract2=((alat+dadct)-x(i))/p(i)
             else
               fract2=(alat-x(i))/p(i)
             endif
           elseif (p(i).lt.0.d0) then
             if (dadct .ge. 0.d0) then
               fract2=-(x(i)-alat)/p(i)
             else
               fract2=-(x(i)-(alat+dadct))/p(i)
             endif
           endif
         endif
         fract=min(fract1,fract2)
         x(i)=x(i)+0.99999d0*p(i)*fract
13       continue
        if ((errx.le.tolx).and.(errf.le.1.d6*tolf)) return
14    continue

      write(6,*) 'maximum number of iterations exceeded in mnewt'
      if (which .eq. 1) write(6,*) 'common tangent problem'
      if (which .eq. 2) write(6,*) 'bessel coefficient problem'
      if (which .eq. 3) write(6,*) 'delta problem'
      if (which .eq. 4) write(6,*) 'alat problem'
      stop
c
```

```
      END


c!!!!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE ENERGY     !!!!!!!!!!!!!!!!!!


      subroutine energy(phn,vm)
c
c calculate pertinent free energies and derivatives
c
      include 'common.h'
      integer phn
c
c     gppt(i)=free energy of phase (i) in units of J/mole
c     ghost = free energy of binary solid solution evaluated at ssconc
c     ghoste(i) = free energy of binary solid solution evaluated at ceqbulk(i)
c     dghost = derivative of ghost w.r.t. composition
c     dghoste(i) = derivative of ghoste(i) w.r.t. composition
c     d2ghost = 2nd derivative of ghost w.r.t. composition
c
c atomic fraction solute in phase phn
c
      cph=1.d0/(stoich(phn)+1.d0)
c
c ibin = 2 -> Al:Sc
c
      if (ibin .eq. 2) then
        a0 = (3347.0d0 -.02d0*temp)
        a1 = (-74918.0d0-11.021d0*temp)
        ghost = a0*ssconc +
     $      a1*ssconc*(1.0d0-ssconc) +
     $      8.314d0*temp*
     $      (ssconc*dlog(ssconc)+(1.0d0-ssconc)*dlog(1.0d0-ssconc))
        ghoste(phn) = a0*ceqbulk(phn) +
     $      a1*ceqbulk(phn)*(1.0d0-ceqbulk(phn)) +
     $      8.314d0*temp*
     $         (ceqbulk(phn)*dlog(ceqbulk(phn)) +
     $      (1.0d0-ceqbulk(phn))*dlog(1.0d0-ceqbulk(phn)))
        dghost = 8.314d0*temp*
     $      ( dlog(ssconc) - dlog(1.0d0-ssconc) ) + a0 +
     $       a1*(1.0d0-2.0d0*ssconc)
        dghoste(phn) = 8.314d0*temp*
     $      ( dlog(ceqbulk(phn)) - dlog(1.0d0-ceqbulk(phn)) ) +
     $      a0 + a1*(1.0d0-2.0d0*ceqbulk(phn))
        d2ghost = 8.314d0*temp/(ssconc*(1.d0-ssconc)) - 2.0d0*a1
c
        gppt(phn)=(-34791.d0 + 2.789d0*temp) +
     $      coh(phn)*vm*avogadro*cph
      else
        write(6,*)'no thermodynamic data for your solute type!!!'
        stop
      endif
c
c calculate delta G(volume) in J/mole
```

```
c
      dgv(phn) = (1.d0-cph) *
     $    ( (ghoste(phn) - ceqbulk(phn)*dghoste(phn)) -
     $      (ghost - ssconc*dghost) ) +
     $    cph *
     $    ( (ghoste(phn) + (1.d0-ceqbulk(phn))*dghoste(phn)) -
     $      (ghost + (1.d0-ssconc)*dghost) )
c
c convert to J/m**3
      dgv(phn) = dgv(phn)/(vm*cph)/avogadro
c
      return
      end


c!!!!!!!!!!!!!!!!!!!!!    SUBROUTINE COMTAN    !!!!!!!!!!!!!!!!!


      subroutine comtan(x,n,np,fvec,fjac,phn,atn,isig)
c
c*******************************************************
c  find common tangent to solid solution free energy and
c  2nd phase free energy by solving the 2 equilibrium conditions that:
c  1)  chem. pot. of A in fcc = chem. pot. of A in 2nd phase
c  2)  same for B
c  - return solid solution and 2nd phase equilibrium concentration "x"
c*******************************************************
c
      include 'common.h'
      integer phn,atn
      dimension x(2),fvec(2),fjac(2,2)
      dimension f(nphas),df(nphas),ddf(nphas)
c
c     gamma(i)=interfacial energy of phase i (J/m^2)
c     coh(i)=coherency strain term (J/m^3)
c     gbe(i)=grain boundary energy (J/m^2)
c     interfarea=grain boundary area
c     f(i)=free energy of phase (i) in units of J/mole
c     df(i)= derivative
c     ddf(i) = 2nd derivative
c     ffcc = free energy of binary solid solution evaluated at x(1)
c     ffcc2 = free energy of binary solid solution evaluated at x(2)
c     dffcc = derivative of ffcc w.r.t. composition
c     ddffcc = 2nd derivative
c
c atomic fraction solute in phase phn
c
      cph=1.d0/(stoich(phn)+1.d0)
c
      if (icoh(phn) .eq. 0) then
        vm = omega(phn)
      elseif (icoh(phn) .eq. 1) then
        vm = volph(phn)
      endif
```

```
c
c ibin = 2 -> Al:Sc
c
      if (ibin .eq. 2) then
        a0 = (3347.0d0 -.02d0*temp)
        a1 = (-74918.0d0-11.021d0*temp)
        ffcc = a0*x(1) +
     $      a1*x(1)*(1.0d0-x(1)) +
     $      8.314d0*temp*
     $          (x(1)*dlog(x(1))+(1.0d0-x(1))*dlog(1.0d0-x(1)))
        ffcc2 = a0*x(2) +
     $      a1*x(2)*(1.0d0-x(2)) +
     $      8.314d0*temp*
     $          (x(2)*dlog(x(2))+(1.0d0-x(2))*dlog(1.0d0-x(2)))
        dffcc = 8.314d0*temp*
     $      ( dlog(x(1)) - dlog(1.0d0-x(1)) ) + a0 +
     $      a1*(1.0d0-2.0d0*x(1))
        ddffcc = 8.314d0*temp/(x(1)*(1.d0-x(1))) - 2.0d0*a1
c
        f(phn)=(-34791.d0 + 2.789d0*temp) +
     $      coh(phn)*vm*avogadro*cph
        df(phn)= 0.0d0
        ddf(phn)= 0.0d0
      else
        write(6,*)'no thermodynamic data for your solute type!!!'
        stop
      endif
c
c if ffcc2 < f(phn) (solid solution is more stable than 2nd phase) then
c   2nd phase cannot form
c
      if (ffcc2 .lt. f(phn) ) then
        isig = 1
        x(1) = cph
        return
      endif
c
c fvec(1) -> delta(chem. pot.)_Al = 0
c fvec(2) -> delta(chem. pot.)_solute = 0
c fjac(i,j) -> jacobian of above 2 eqns
c
c if 2nd phase is stoichiometric, only 1 equation and 1 unknown composition
c so make 2nd equation pin stoichiometric composition
c
      if (atn .eq. natom) then
        cap = 0.d0
      else
c         cap = (gamma(phn)*area2(atn,phn)-gbe(phn)*interfarea2(atn,phn))
c    $        *avogadro*cph
        cap = (gamma(phn)*area2(atn,phn))*avogadro*cph
      endif
      if (df(phn) .eq. 0.0d0) then
        n = 1
        fvec(1) = -dffcc*(x(1)-cph) + ffcc - f(phn) - cap
```

```
          fjac(1,1) = -ddffcc*(x(1)-cph)
        else
          n = 2
          fvec(1) = -dffcc*x(1) + ffcc + df(phn)*x(2) - f(phn) - cap
          fvec(2) = dffcc*(1.0d0-x(1)) + ffcc
     $       - df(phn)*(1.0d0-x(2)) - f(phn) - cap
          fjac(1,1) = -ddffcc*x(1)
          fjac(1,2) = ddf(phn)*x(2)
          fjac(2,1) = ddffcc*(1.0d0-x(1))
          fjac(2,2) = -ddf(phn)*(1.0d0-x(2))
        endif
c
        return
        end


c!!!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE GETDELT   !!!!!!!!!!!!!!!!!


        subroutine getdelt(x,n,np,fvec,fjac,phn)
c
c get delta for n* + delta
c
        include 'common.h'
        integer phn
        dimension x(1),fvec(1),fjac(1,1)
c
        n = 1
c
        rt = 8.314d0*temp
c
        fvec(1) = x(1)*x(1)*x(1) + (1.5d0*x(1)*x(1)) -
     $    (3.d0*rt/(avogadro*8.d0*pi*gamma(phn)*rstar(phn)*rstar(phn)))
c
        fjac(1,1) = 3.d0*x(1)*x(1) + 3.d0*x(1)
c
        return
        end


c!!!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE GETALAT   !!!!!!!!!!!!!!!!!


        subroutine getalat(x,n,np,fvec,fjac)
c
c get solid solution concentration dependent alat
c
        include 'common.h'
        dimension x(1),fvec(1),fjac(1,1)
c
        n = 1
c
        x34 = 0.25d0 * x(1) * x(1) * x(1)
c
```

```
c     fvec(1) = x(1) - alat - (x34 * sdadc) /
c    $     (1.d0 - tvincp - (x34 * tvcohp))
      fvec(1) = (x(1)-alat)*(1.d0 - tvincp - (x34 * tvcohp)) -
     $     (x34 * sdadc)
c
c      fjac(1,1) = 1.d0 - (0.75d0 * x(1)*x(1) * sdadc) *
c    $     ( (1.d0 - tvincp - (2.d0 * x34 * tvcohp)) /
c    $       (1.d0 - tvincp - (x34 * tvcohp))**2 )
      fjac(1,1) = (1.d0 - tvincp - (x34 * tvcohp)) -
     $     (x(1)-alat)*(0.75d0 * x(1)*x(1) *tvcohp) -
     $     (0.75d0 * x(1)*x(1) * sdadc)
c
      return
      end


c!!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE LUDCMP     !!!!!!!!!!!!!!!!


      SUBROUTINE ludcmp(a,n,np,indx,d)
      INTEGER n,np,indx(n),NMAX
      REAL*8 d,a(np,np),TINY
      PARAMETER (NMAX=500,TINY=1.0e-20)
      INTEGER i,imax,j,k
      REAL*8 aamax,dum,sum,vv(NMAX)
      d=1.
      do 12 i=1,n
        aamax=0.
        do 11 j=1,n
          if (abs(a(i,j)).gt.aamax) aamax=abs(a(i,j))
11      continue
        if (aamax.eq.0.) pause 'singular matrix in ludcmp'
        vv(i)=1./aamax
12    continue
      do 19 j=1,n
        do 14 i=1,j-1
          sum=a(i,j)
          do 13 k=1,i-1
            sum=sum-a(i,k)*a(k,j)
13        continue
          a(i,j)=sum
14      continue
        aamax=0.
        do 16 i=j,n
          sum=a(i,j)
          do 15 k=1,j-1
            sum=sum-a(i,k)*a(k,j)
15        continue
          a(i,j)=sum
          dum=vv(i)*abs(sum)
          if (dum.ge.aamax) then
            imax=i
            aamax=dum
          endif
```

```
16     continue
      if (j.ne.imax)then
        do 17 k=1,n
          dum=a(imax,k)
          a(imax,k)=a(j,k)
          a(j,k)=dum
17        continue
        d=-d
        vv(imax)=vv(j)
      endif
      indx(j)=imax
       if(a(j,j).eq.0.)a(j,j)=TINY
       if(j.ne.n)then
        dum=1./a(j,j)
        do 18 i=j+1,n
          a(i,j)=a(i,j)*dum
18        continue
      endif
19    continue
     return
     END


c!!!!!!!!!!!!!!!!!!!!!!     SUBROUTINE LUBKSB     !!!!!!!!!!!!!!!!!


      SUBROUTINE lubksb(a,n,np,indx,b)
     INTEGER n,np,indx(n)
     REAL*8 a(np,np),b(n)
     INTEGER i,ii,j,ll
     REAL*8 sum
     ii=0
     do 12 i=1,n
       ll=indx(i)
       sum=b(ll)
       b(ll)=b(i)
       if (ii.ne.0)then
         do 11 j=ii,i-1
           sum=sum-a(i,j)*b(j)
11         continue
       else if (sum.ne.0.d0) then
         ii=i
       endif
       b(i)=sum
12    continue
      do 14 i=n,1,-1
       sum=b(i)
       do 13 j=i+1,n
         sum=sum-a(i,j)*b(j)
13       continue
       b(i)=sum/a(i,i)
14    continue
     return
     END
```

```
c!!!!!!!!!!!!!!!!!!!!    COMMON FILE 'common.h'    !!!!!!!!!!!!!!!!!!!!

      implicit real*8 (a-h,o-z)
      real*8 maxtime,maxtau
c if following parameter values are changed, change NMAX in rkqs and rkck
      parameter(natom=100000,nphas=2)
      parameter(natph=natom*nphas)
      parameter(nvac=1)
      parameter(avogadro=6.02217d23)
c
      integer critsize,critmin,driftcut
      real*8 interfarea,interfarea2
c
c  dadc (m/mole fraction) specified in the following order:
c     Mg, Sc, Ag, Cu, Cd,...
c
      dimension dadc(10)
       data dadc/0.00000000,+0.480d-12,0.00000000,0.00000000,
     $    0.00000000,
     $     0.0d-10,0.0d-10,0.0d-10,0.0d-10,0.0d-10/
c
      common/nph/nph,nat(nphas)
      common/inmat/alat,radres,difc,em,e2m,volph(nphas),
     $          ev,difvac,evacm,difdivac,edivacm,
     $          sconf(3),evacb(3),
     $          eccen(nphas),dibind(nphas),
     $          gamma(nphas),coh(nphas),gbe(nphas),
     $          rhod,stoich(nphas),gthick,gdiam,tconc,ibin,nans,
     $          icoh(nphas)
      common/int/tinit,tfin,corat
      common/initer/tau,tautest,time,timeinit,maxtime,temp,dcon,
     $          dvac,ddivac,tempold,nit,imethod
      common/hmax/maxtau
      common/volume/rad(natom+1,nphas),area(natom+1,nphas),
     $          volume(natom+1,nphas),interfarea(natom+1,nphas),
     $          omega(nphas),vol,alatss,
     $          area2(natom+1,nphas),vol2(natom+1,nphas),
     $          interfarea2(natom+1,nphas)
      common/concentration/c0(natom,nphas),ceq(natom,nphas),
     $          ceqbulk(nphas),ceqgt(natom,nphas),
     $          precipconc(nphas),sumc0(nphas),chk,
     $          ssconc
      common/rates/grow(natom,nphas),emit(natom,nphas),
     $          flux(natom,nphas),
     $          growvac(nvac),emitvac(nvac),vacflarea(nvac)
      common/freeenergy/ghost,ghoste(nphas),gppt(nphas),
     $          dghost,dghoste(nphas),dgv(nphas),d2ghost
      common/vac/cvac(nvac),cveq,rvac(nvac),rdist,chkvac
      common/size/avsz(nphas,3),avsztot(nphas),totc0(nphas,3),
     $          rstar(nphas),critsize(nphas),critmin(nphas),
     $          driftcut(nphas),
     $          eobl1(nphas),eobl2(nphas),epro1(nphas),epro2(nphas)
      common/ssalat/tvincp,tvcohp,sdadc
      common/trig/pi
      common/cuberoots/cbrts(0:natom),delsqrt(nvac)
```

Sample input file

```
 $inpcard
  tinit = 923.0d0,
  tfin = 598.0d0,
  corat = 2000.d0,
  tconc = .0011d0,
  tau = 1.d-15,
  maxtau = 1.d0,
  nit = 1000000,
  tautest = 2.d0,
  imethod = 1,
  maxtime = 3600.d0,
  nph = 1,
  stoich(1) = 3.0d0,
  alat = 4.05d-10,
  volph(1) = 69.224d-30,
  eccen(1) = 1.d0,
  icoh(1) = 1,
  gamma(1) = .075d0,
  coh(1) = 1.917d+7,
  gbe(1) = 0.0d0,
  gthick = 0.d0,
  gdiam = 1.d-6,
  difc =  3560.d0,
  em = 2.318d0,
  e2m = 999.d0,
  ev = .66d0,
  difvac = .13d-4,
  evacm = 0.62d0,
  difdivac = 0.d0,
  edivacm = 999.d0,
  evacb = 0.0d0,
  sconf = 1.d0,
  rhod = 1.d10,
  ibin = 2,
  dibind = 0.129d0,
  nans = 0,
  resol=10.d-10
$end
```

UNLIMITED RELEASE

INITIAL DISTRIBUTION

Professor D. de Fontaine
Department of Materials Science and
Mineral Engineering
University of California at Berkeley
Berkeley, CA 94720

Professor J. J. Hoyt
School of Mechanical and Materials Engineering
Washington State University
Pullman, WA 99164

Robert W. Hyland, Jr.
Alcoa Technical Center, Aluminum Company
of America
100 Technical Drive
Alcoa Center, PA 15069

Cathy Rohrer
Alcoa Technical Center, Aluminum Company
of America
100 Technical Drive
Alcoa Center, PA 15069

Barbara Hall
Alcoa Technical Center, Aluminum Company
of America
100 Technical Drive
Alcoa Center, PA 15069

Joanne Murray
Alcoa Technical Center, Aluminum Company
of America
100 Technical Drive
Alcoa Center, PA 15069

Don Ziegler
Alcoa Technical Center, Aluminum Company
of America
100 Technical Drive
Alcoa Center, PA 15069

Ken Lippert
Alcoa Technical Center, Aluminum Company
of America
100 Technical Drive
Alcoa Center, PA 15069

MS 0342 Dick Salzbrenner, 1805
MS 0342 Kim W. Mahin, 1807
MS 1380 Dave Larson, 4231 (6)
MS 1411 Elizabeth A. Holm, 1841
MS 1411 Thomas Swiler, 1841
MS 1803 James L. Jellison, 1803
MS 9001 T. O. Hunter, 8000
MS 9401 T. M. Dyer, 8700
MS 9161 W. G. Wolfer, 8717
MS 9161 Mark Asta, 8717 (50)
MS 9402 Mark Perra, 8711
MS 9403 Michael Baskes, 8712
MS 9430 Anton J. West, 8240
MS 9021 Technical Communications Department, 8815, for OSTI (10)
MS 9021 Technical Communications Department, 8815
MS 0899 Technical Library, 4414 (4)
MS 9018 Central Technical Files, 8940-2 (3)